

## Template Documentation



# Welcome to the Template Project Documentation

This is the template project with all the required libraries

## Getting Started

To get started, add a documentation source to the project (a Visual Studio solution, project, or assembly and XML comments file). See the **Getting Started** topics in the Sandcastle Help File Builder's help file for more information. The following default items are included in this project:

- *ContentLayout.content* - Use the content layout file to manage the conceptual content in the project and define its layout in the table of contents.
- The *.lmedia* folder - Place images in this folder that you will reference from conceptual content using `mediaLink` or `mediaLinkInline` elements. If you will not have any images in the file, you may remove this folder.
- The *.licons* folder - This contains a default logo for the help file. You may replace it or remove it and the folder if not wanted. If removed or if you change the file name, update the **Transform Args** project properties page by removing or changing the filename in the `LogoFile` transform argument. Note that unlike images referenced from conceptual topics, the logo file should have its **BuildAction** property set to `Content`.
- The *.lContent* folder - Use this to store your conceptual topics. You may name the files and organize them however you like. One suggestion is to lay the files out on disk as you have them in the content layout file as shown in this project but the choice is yours. Files can be added via the Solution Explorer or from within the content layout file editor. Files must appear in the content layout file in order to be compiled into the help file.

See the **Conceptual Content** topics in the Sandcastle Help File Builder's help file for more information. See the **Sandcastle MAML**

**Guide** for details on Microsoft Assistance Markup Language (MAML) which is used to create these topics.

## See Also

[Other Resources](#)

[Version History](#)

---

## Template Documentation



# Version History

The topics in this section describe the various changes made to the [TODO: Project Title] over the life of the project.

## ▪ Version History

Select a version below to see a description of its changes.

- [Version 1.0.0.0](#)
- [TODO: Add links to each specific version page]

## ▪ See Also

Other Resources

[Welcome to the Template Project Documentation](#)

---

## Template Documentation



# Version 1.0.0.0

Version 1.0.0.0 was released on 23/05/2017.

## ▪ Changes in This Release

- [TODO: Add change items here]

## ▪ See Also

Other Resources

[Version History](#)

---

## Template Documentation



# Template.Core.Interface Namespace

## ▪ Interfaces

Interface	Description
<a href="#">IGenericRepositoryTEntity</a>	Declares all the funtions with the most commons database actions

## Template Documentation



# IGenericRepository<TEntity> Interface

Declares all the funtions with the most commons database actions

**Namespace:** [Template.Core.Interface](#)

**Assembly:** Template.Repository (in Template.Repository.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

C#

[Copy](#)

```
public interface IGenericRepository<TEntity>
where TEntity : BaseEntity
```

### Type Parameters

*TEntity*

The Entity Framework entity that represents a table

The [IGenericRepository<TEntity>](#) type exposes the following members.

## ► Methods

	Name	Description
	<a href="#">Add</a>	Add a new entity to the context, it will be added to the database on Save
	<a href="#">AddIEnumerable</a>	Add a IEnumerable of new entities to the context, they will be added to the

database on Save

 <a href="#">Delete</a>	Marks the given entity as deleted, the entity will be deleted on Save
 <a href="#">DeleteBy</a>	Marks the entities that match with the given predicate as deleted, they will be deleted of the database on Save
 <a href="#">Edit</a>	Marks the passed entity as Modified, the changes will persist on Save
 <a href="#">FindBy</a>	Filters the results by the passed predicate
 <a href="#">Get</a>	Gets an unique TEntity from the database
 <a href="#">GetAll</a>	Get all the results of the IEntity
 <a href="#">Save</a>	Save all the changes made to the context
 <a href="#">SelectWhereTResult</a>	This function allows to select the specific properties and filter by where clause on database, not on memory, in that way it improves the performance

[Top](#)

## See Also

## Reference

### Template.Core.Interface Namespace

---

## Template Documentation



# IGenericRepository Methods

The [IGenericRepository](#) generic type exposes the following members.

## Methods

	Name	Description
	<a href="#">Add</a>	Add a new entity to the context, it will be added to the database on Save
	<a href="#">AddIEnumerable</a>	Add a IEnumerable of new entities to the context, they will be added to the database on Save
	<a href="#">Delete</a>	Marks the given entity as deleted, the entity will be deleted on Save
	<a href="#">DeleteBy</a>	Marks the entities that match with the given predicate as deleted, they will be deleted of the database on Save
	<a href="#">Edit</a>	Marks the passed entity as Modified, the changes will persist on Save
	<a href="#">FindBy</a>	Filters the results by the

		passed predicate
≡	<a href="#">Get</a>	Gets an unique TEntity from the database
≡	<a href="#">GetAll</a>	Get all the results of the IEntity
≡	<a href="#">Save</a>	Save all the changes made to the context
≡	<a href="#">SelectWhereTResult</a>	This function allows to select the specific properties and filter by where clause on database, not on memory, in that way it improves the performance

[Top](#)

## ▲ See Also

Reference

[IGenericRepository Interface](#)  
[Template.Core.Interface Namespace](#)

## Template Documentation



# IGenericRepository{TEntity} Add Method

Add a new entity to the context, it will be added to the database on Save

**Namespace:** [Template.Core.Interface](#)

**Assembly:** Template.Repository (in Template.Repository.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

C#

[Copy](#)

```
TEntity Add(  
            TEntity entity  
)
```

### Parameters

*entity*

Type: [TEntity](#)

The entity to be added

### Return Value

Type: [TEntity](#)

The entity after being added

## ► See Also

### Reference

[IGenericRepository{TEntity} Interface](#)  
[Template.Core.Interface Namespace](#)

## Template Documentation



# IGenericRepository<TEntity>.AddIEnum Method

Add a IEnumerable of new entities to the context, they will be added to the database on Save

**Namespace:** [Template.Core.Interface](#)

**Assembly:** Template.Repository (in Template.Repository.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

C#

[Copy](#)

```
IEnumerable<TEntity> AddIEnumerable(  
    IEnumerable<TEntity> entities  
)
```

### Parameters

*entities*

Type: [System.Collections.Generic.IEnumerable<TEntity>](#)

The list of entities to insert

### Return Value

Type: [IEnumerable<TEntity>](#)

The IEnumerable of entities after being added

## ► See Also

### Reference

[IGenericRepository<TEntity> Interface](#)

[Template.Core.Interface Namespace](#)

## Template Documentation



# IGenericRepository{TEntity}Delete Method

Marks the given entity as deleted, the entity will be deleted on Save

**Namespace:** [Template.Core.Interface](#)

**Assembly:** Template.Repository (in Template.Repository.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

```
C#  
TEntity Delete(  
    TEntity entity  
)
```

[Copy](#)

### Parameters

*entity*

Type: [TEntity](#)

The entity to be removed

### Return Value

Type: [TEntity](#)

The deleted entity

## ► See Also

Reference

[IGenericRepository{TEntity} Interface](#)

[Template.Core.Interface Namespace](#)

## Template Documentation



# IGenericRepository TEntityDeleteBy Method

Marks the entities that match with the given predicate as deleted, they will be deleted of the database on Save

**Namespace:** [Template.Core.Interface](#)

**Assembly:** Template.Repository (in Template.Repository.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

```
C# Copy
int DeleteBy(
    Expression<Func< TEntity, bool>> predicate
)
```

### Parameters

*predicate*

Type: [System.Linq.Expressions.Expression<Func< TEntity, Boolean>>](#)  
The predicate to filter the result

### Return Value

Type: [Int32](#)

The number of affected rows entity

## ► See Also

### Reference

[IGenericRepository TEntity Interface](#)  
[Template.Core.Interface Namespace](#)



## Template Documentation



# IGenericRepositoryEdit Method

Marks the passed entity as Modified, the changes will persist on Save

**Namespace:** [Template.Core.Interface](#)

**Assembly:** Template.Repository (in Template.Repository.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

C#

Copy

```
TEntity Edit(  
    TEntity entity  
)
```

### Parameters

*entity*

Type: [TEntity](#)

[Missing <param name="entity"/> documentation for  
"M:Template.Core.Interface.IGenericRepository`1.Edit(`0)"]

### Return Value

Type: [TEntity](#)

The modified entity

## ► See Also

### Reference

[IGenericRepository Interface](#)  
[Template.Core.Interface Namespace](#)

## Template Documentation



# IGenericRepositoryFindBy Method

Filters the results by the passed predicate

**Namespace:** [Template.Core.Interface](#)

**Assembly:** Template.Repository (in Template.Repository.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

```
C# Copy
IEnumerable< TEntity > FindBy(
    Expression< Func< TEntity, bool > > predicate
)
```

### Parameters

*predicate*

Type: [System.Linq.Expressions.Expression<Func< TEntity, Boolean >>](#)  
The expression to filter the result

### Return Value

Type: [IEnumerable< TEntity >](#)

The IEnumerable with the results

## ► See Also

### Reference

[IGenericRepository Interface](#)  
[Template.Core.Interface Namespace](#)

## Template Documentation



# IGenericRepositoryGet Method

Gets an unique TEntity from the database

**Namespace:** [Template.Core.Interface](#)

**Assembly:** Template.Repository (in Template.Repository.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

```
C# Copy
    TEntity Get(
        Expression<Func<
```

### Parameters

*predicate*

Type: [System.Linq.Expressions.Expression<Func< TEntity, Boolean >>](#)  
The predicate to filter the result

### Return Value

Type: [TEntity](#)

The TEntity of the result

## ► See Also

### Reference

[IGenericRepository Interface](#)  
[Template.Core.Interface Namespace](#)

## Template Documentation



# IGenericRepository GetAll Method

Get all the results of the IEntity

**Namespace:** [Template.Core.Interface](#)

**Assembly:** Template.Repository (in Template.Repository.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

C#

[Copy](#)

```
IEnumerable< TEntity > GetAll()
```

Return Value

Type: [IEnumerable](#)

returns an IEnumerable with all the results

## ► See Also

Reference

[IGenericRepository Interface](#)

[Template.Core.Interface Namespace](#)

## Template Documentation



# IGenericRepository Save Method

Save all the changes made to the context

**Namespace:** [Template.Core.Interface](#)

**Assembly:** Template.Repository (in Template.Repository.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

C#

```
int Save()
```

[Copy](#)

Return Value

Type: [Int32](#)

The number of affected rows

## ► See Also

Reference

[IGenericRepository Interface](#)

[Template.Core.Interface Namespace](#)

## Template Documentation



# IGenericRepositorySelectWhere Method

This function allows to select the specific properties and filter by where clause on database, not on memory, in that way it improves the performance

**Namespace:** [Template.Core.Interface](#)

**Assembly:** Template.Repository (in Template.Repository.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

```
C# Copy
IEnumerable<TResult> SelectWhere<TResult>(
    Func< TEntity, TResult > selector,
    Expression<Func< TEntity, bool >> predicate
)
```

## Parameters

*selector*

Type: [System.Func< TEntity, TResult >](#)

The properties to select

*predicate*

Type: [System.Linq.Expressions.Expression<Func< TEntity, Boolean >>](#)

The where clause

## Type Parameters

*TResult*

The select result type

**Return Value**

Type: [IEnumerable<TResult>](#)

The IEnumerable of the result

## See Also

**Reference**

[IGenericRepository< TEntity > Interface](#)

[Template.Core.Interface Namespace](#)

---

## Template Documentation



# Template.Repository.Class Namespace

## ► Classes

Class	Description
 <a href="#">MoviesCategoryRepository</a>	A class repository to manage all the user profile actions
 <a href="#">MoviesRepository</a>	A class repository to manage all the movies category actions

## Template Documentation



# MoviesCategoryRepository Class

A class repository to manage all the user profile actions

## ► Inheritance Hierarchy

[SystemObject](#) [Template.Repository.CoreGenericRepository](#)[MOVIE\\_C](#)  
[Template.Repository.ClassMoviesCategoryRepository](#)

**Namespace:** [Template.Repository.Class](#)

**Assembly:** [Template.Repository](#) (in [Template.Repository.dll](#)) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

```
C# Copy
public class MoviesCategoryRepository : GenericRe
    IMoviesCategoryRepository, IGenericRe
```

The [MoviesCategoryRepository](#) type exposes the following members.

## ► Constructors

Name	Description
<a href="#">MoviesCategoryRepository</a>	Constructor of <a href="#">MoviesCategoryRepository</a>

[Top](#)

## ► Properties

Name	Description
 <a href="#">_entities</a>	

[Top](#)

## Methods

Name	Description
 <a href="#">Add</a>	Add a new entity to the context, it will be added to the database on Save (Inherited from <a href="#">GenericRepositoryTEntity</a> .)
 <a href="#">AddIEnumerable</a>	Add a IEnumerable of new entities to the context, they will be added to the database on Save (Inherited from <a href="#">GenericRepositoryTEntity</a> .)
 <a href="#">Delete</a>	Marks the given entity as deleted, the entity will be deleted on Save (Inherited from <a href="#">GenericRepositoryTEntity</a> .)
 <a href="#">DeleteBy</a>	Marks the entities that match with the given predicate as deleted, they will be deleted of the database on Save (Inherited from <a href="#">GenericRepositoryTEntity</a> .)
 <a href="#">Edit</a>	Marks the passed entity as Modified, the changes will persist

on Save  
(Inherited from  
[GenericRepositoryTEntity](#).)

≡	<a href="#">FindBy</a>	Filters the results by the passed predicate (Inherited from <a href="#">GenericRepositoryTEntity</a> .)
≡	<a href="#">Get</a>	Gets an unique TEntity from the database (Inherited from <a href="#">GenericRepositoryTEntity</a> .)
≡	<a href="#">GetAll</a>	Retrieves all the UserProfiles entities with their roles (Overrides <a href="#">GenericRepositoryTEntity.GetAll</a> .)
≡	<a href="#">Save</a>	Save all the changes made to the context (Inherited from <a href="#">GenericRepositoryTEntity</a> .)
≡	<a href="#">SelectWhereTResult</a>	This function allows to select the specific properties and filter by where clause on database, not on memory, in that way it improves the performance (Inherited from <a href="#">GenericRepositoryTEntity</a> .)

[Top](#)

## Fields

Name	Description
------	-------------



[\\_DBEntities](#) The DbContext of entity framework  
(Inherited from  
[GenericRepositoryTEntity](#).)

---



[\\_Entity](#) The DbSet that represents the database table to do all the queries  
(Inherited from  
[GenericRepositoryTEntity](#).)

---

[Top](#)

## ▲ See Also

Reference

[Template.Repository.Class Namespace](#)

## Template Documentation



# MoviesCategoryRepository Constructor

Constructor of [MoviesCategoryRepository](#)

**Namespace:** [Template.Repository.Class](#)

**Assembly:** Template.Repository (in Template.Repository.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

C#

[Copy](#)

```
public MoviesCategoryRepository(  
    DbContext entities  
)
```

## Parameters

*entities*

Type: [DbContext](#)

The DbContext to do the transaction injected by UnitOfWork

## ► See Also

### Reference

[MoviesCategoryRepository Class](#)  
[Template.Repository.Class Namespace](#)

## Template Documentation



# MoviesCategoryRepository Properties

## ► Properties

Name	Description
 _entities	

[Top](#)

## ► See Also

[Reference](#)

[MoviesCategoryRepository Class](#)

[Template.Repository.Class Namespace](#)

## Template Documentation



# MoviesCategoryRepository\_entities Property

[Missing <summary> documentation for  
"P:Template.Repository.Class.MoviesCategoryRepository.\_entities"]

**Namespace:** [Template.Repository.Class](#)

**Assembly:** Template.Repository (in Template.Repository.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

C#

[Copy](#)

```
private ExampleMoviesEntities _entities { get; }
```

Property Value

Type: [ExampleMoviesEntities](#)

## ► See Also

[Reference](#)

[MoviesCategoryRepository Class](#)

[Template.Repository.Class Namespace](#)

## Template Documentation



# MoviesCategoryRepository Methods

The [MoviesCategoryRepository](#) type exposes the following members.

## ▪ Methods

Name	Description
<a href="#">Add</a>	Add a new entity to the context, it will be added to the database on Save (Inherited from <a href="#">GenericRepositoryTEntity</a> .)
<a href="#">AddIEnumerable</a>	Add a IEnumerable of new entities to the context, they will be added to the database on Save (Inherited from <a href="#">GenericRepositoryTEntity</a> .)
<a href="#">Delete</a>	Marks the given entity as deleted, the entity will be deleted on Save (Inherited from <a href="#">GenericRepositoryTEntity</a> .)
<a href="#">DeleteBy</a>	Marks the entities that match with the given predicate as deleted, they will be deleted of the database on Save (Inherited from <a href="#">GenericRepositoryTEntity</a> .)

---

≡•	<a href="#">Edit</a>	Marks the passed entity as Modified, the changes will persist on Save (Inherited from <a href="#">GenericRepositoryTEntity</a> .)
≡•	<a href="#">FindBy</a>	Filters the results by the passed predicate (Inherited from <a href="#">GenericRepositoryTEntity</a> .)
≡•	<a href="#">Get</a>	Gets an unique TEntity from the database (Inherited from <a href="#">GenericRepositoryTEntity</a> .)
≡•	<a href="#">GetAll</a>	Retrieves all the UserProfiles entities with their roles (Overrides <a href="#">GenericRepositoryTEntity.GetAll</a> .)
≡•	<a href="#">Save</a>	Save all the changes made to the context (Inherited from <a href="#">GenericRepositoryTEntity</a> .)
≡•	<a href="#">SelectWhereTResult</a>	This function allows to select the specific properties and filter by where clause on database, not on memory, in that way it improves the performance (Inherited from <a href="#">GenericRepositoryTEntity</a> .)

---

[Top](#)

## ▲ See Also

Reference

[MoviesCategoryRepository Class](#)

[Template.Repository.Class Namespace](#)

---

## Template Documentation



# MoviesCategoryRepository GetAll Method

Retrieves all the UserProfiles entities with their roles

**Namespace:** [Template.Repository.Class](#)

**Assembly:** Template.Repository (in Template.Repository.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

```
C# Copy
public override IEnumerable<MOVIE_CATEGORY> GetAll()
```

**Return Value**

Type: [IEnumerableMOVIE\\_CATEGORY](#)

An IEnumerable of UserProfile

**Implements**

[IGenericRepositoryTEntityGetAll](#)

[IGenericRepositoryTEntityGetAll](#)

## ► See Also

**Reference**

[MoviesCategoryRepository Class](#)

[Template.Repository.Class Namespace](#)

## Template Documentation



# MoviesCategoryRepository Fields

The [MoviesCategoryRepository](#) type exposes the following members.

## ▪ Fields

	Name	Description
💡	<a href="#">_DBEntities</a>	The DbContext of entity framework (Inherited from <a href="#">GenericRepositoryTEntity</a> .)
💡	<a href="#">_Entity</a>	The DbSet that represents the database table to do all the queries (Inherited from <a href="#">GenericRepositoryTEntity</a> .)

[Top](#)

## ▪ See Also

Reference

[MoviesCategoryRepository Class](#)  
[Template.Repository.Class Namespace](#)

## Template Documentation



# MoviesRepository Class

A class repository to manage all the movies category actions

## ► Inheritance Hierarchy

SystemObject Template.Repository.CoreGenericRepositoryMOVIES  
Template.Repository.ClassMoviesRepository

**Namespace:** [Template.Repository.Class](#)

**Assembly:** Template.Repository (in Template.Repository.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

```
C# Copy
public class MoviesRepository : GenericRepository<IMoviesRepository, IGenericRepository<MO\
```

The [MoviesRepository](#) type exposes the following members.

## ► Constructors

Name	Description
<a href="#">MoviesRepository</a>	Constructor of <a href="#">MoviesCategoryRepository</a>

[Top](#)

## ► Properties

Name	Description



## \_entities

[Top](#)

# Methods

	Name	Description
≡	<a href="#">Add</a>	Add a new entity to the context, it will be added to the database on Save (Inherited from <a href="#">GenericRepository TEntity</a> .)
≡	<a href="#">AddIEnumerable</a>	Add a IEnumerable of new entities to the context, they will be added to the database on Save (Inherited from <a href="#">GenericRepository TEntity</a> .)
≡	<a href="#">Delete</a>	Marks the given entity as deleted, the entity will be deleted on Save (Inherited from <a href="#">GenericRepository TEntity</a> .)
≡	<a href="#">DeleteBy</a>	Marks the entities that match with the given predicate as deleted, they will be deleted of the database on Save (Inherited from <a href="#">GenericRepository TEntity</a> .)
≡	<a href="#">Edit</a>	Marks the passed entity as Modified, the changes will persist on Save

		(Inherited from <a href="#">GenericRepositoryTEntity</a> .)
≡	<a href="#">FindBy</a>	Filters the results by the passed predicate (Inherited from <a href="#">GenericRepositoryTEntity</a> .)
≡	<a href="#">Get</a>	Gets an unique TEntity from the database (Inherited from <a href="#">GenericRepositoryTEntity</a> .)
≡	<a href="#">GetAll</a>	Get all the results of the IEntity (Inherited from <a href="#">GenericRepositoryTEntity</a> .)
≡	<a href="#">Save</a>	Save all the changes made to the context (Inherited from <a href="#">GenericRepositoryTEntity</a> .)
≡	<a href="#">SelectWhereTResult</a>	This function allows to select the specific properties and filter by where clause on database, not on memory, in that way it improves the performance (Inherited from <a href="#">GenericRepositoryTEntity</a> .)

[Top](#)

## Fields

Name	Description
------	-------------



[\\_DBEntities](#) The DbContext of entity framework  
(Inherited from  
[GenericRepositoryTEntity](#).)

---



[\\_Entity](#) The DbSet that represents the database table to do all the queries  
(Inherited from  
[GenericRepositoryTEntity](#).)

---

[Top](#)

## ▲ See Also

Reference

[Template.Repository.Class Namespace](#)

## Template Documentation



# MoviesRepository Constructor

Constructor of [MoviesCategoryRepository](#)

**Namespace:** [Template.Repository.Class](#)

**Assembly:** Template.Repository (in Template.Repository.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

C#

Copy

```
public MoviesRepository(  
    DbContext entities  
)
```

### Parameters

*entities*

Type: [DbContext](#)

The DbContext to do the transaction injected by UnitOfWork

## ► See Also

Reference

[MoviesRepository Class](#)

[Template.Repository.Class Namespace](#)

## Template Documentation



# MoviesRepository Properties

## ► Properties

Name	Description
 _entities	

[Top](#)

## ► See Also

Reference

[MoviesRepository Class](#)

[Template.Repository.Class Namespace](#)

## Template Documentation



# MoviesRepository\_entities Property

[Missing <summary> documentation for  
"P:Template.Repository.Class.MoviesRepository.\_entities"]

**Namespace:** [Template.Repository.Class](#)

**Assembly:** Template.Repository (in Template.Repository.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

C#

[Copy](#)

```
private ExampleMoviesEntities _entities { get; }
```

Property Value

Type: [ExampleMoviesEntities](#)

## ► See Also

[Reference](#)

[MoviesRepository Class](#)

[Template.Repository.Class Namespace](#)

## Template Documentation



# MoviesRepository Methods

The [MoviesRepository](#) type exposes the following members.

## ▪ Methods

Name	Description
<a href="#">Add</a>	Add a new entity to the context, it will be added to the database on Save (Inherited from <a href="#">GenericRepositoryTEntity</a> .)
<a href="#">AddIEnumerable</a>	Add a IEnumerable of new entities to the context, they will be added to the database on Save (Inherited from <a href="#">GenericRepositoryTEntity</a> .)
<a href="#">Delete</a>	Marks the given entity as deleted, the entity will be deleted on Save (Inherited from <a href="#">GenericRepositoryTEntity</a> .)
<a href="#">DeleteBy</a>	Marks the entities that match with the given predicate as deleted, they will be deleted of the database on Save (Inherited from <a href="#">GenericRepositoryTEntity</a> .)

≡	Edit	Marks the passed entity as Modified, the changes will persist on Save (Inherited from <a href="#">GenericRepositoryTEntity</a> .)
≡	FindBy	Filters the results by the passed predicate (Inherited from <a href="#">GenericRepositoryTEntity</a> .)
≡	Get	Gets an unique TEntity from the database (Inherited from <a href="#">GenericRepositoryTEntity</a> .)
≡	GetAll	Get all the results of the IEntity (Inherited from <a href="#">GenericRepositoryTEntity</a> .)
≡	Save	Save all the changes made to the context (Inherited from <a href="#">GenericRepositoryTEntity</a> .)
≡	SelectWhereTResult	This function allows to select the specific properties and filter by where clause on database, not on memory, in that way it improves the performance (Inherited from <a href="#">GenericRepositoryTEntity</a> .)

[Top](#)

## ▲ See Also

Reference

[MoviesRepository Class](#)

[Template.Repository.Class Namespace](#)

---

## Template Documentation



# MoviesRepository Fields

The [MoviesRepository](#) type exposes the following members.

## ► Fields

	Name	Description
💡	<a href="#">_DBEntities</a>	The DbContext of entity framework (Inherited from <a href="#">GenericRepositoryTEntity</a> .)
💡	<a href="#">_Entity</a>	The DbSet that represents the database table to do all the queries (Inherited from <a href="#">GenericRepositoryTEntity</a> .)

[Top](#)

## ► See Also

Reference

[MoviesRepository Class](#)

[Template.Repository.Class Namespace](#)

## Template Documentation



# Template.Repository.Context Namespace

## ► Classes

	Class	Description
	<a href="#">ExampleMoviesEntities</a>	The DbContext class of Entity Framework
	<a href="#">MOVIE_CATEGORY</a>	Generated entity class by Entity Framework
	<a href="#">MOVIES</a>	Generated entity class by Entity Framework

## Template Documentation



# ExampleMoviesEntities Class

The DbContext class of Entity Framework

## ▪ Inheritance Hierarchy

```
SystemObject DbContext  
Template.Repository.ContextExampleMoviesEntities
```

**Namespace:** [Template.Repository.Context](#)

**Assembly:** Template.Repository (in Template.Repository.dll) Version: 1.0.0.0 (1.0.0.0)

## ▪ Syntax

C#

Copy

```
public class ExampleMoviesEntities : DbContext
```

The [ExampleMoviesEntities](#) type exposes the following members.

## ▪ Constructors

	Name	Description
≡	<a href="#">ExampleMoviesEntities</a>	The default constructor for the Entity class

[Top](#)

## ▪ Properties

	Name	Description
	<a href="#">MOVIE_CATEGORY</a>	Represents the table

MOVIE\_CATEGORY of the database with the base CRUD functions



## MOVIES

Represents the table MOVIES of the database with the base CRUD functions

[Top](#)

## Methods

	Name	Description
	<a href="#">OnModelCreating</a>	Overrides the base function to only throw exception when is called, because this function is only called on a CodeFirst solution (Overrides <a href="#">DbContext.OnModelCreating(DbModelBuilder)</a> )

[Top](#)

## See Also

Reference

[Template.Repository.Context Namespace](#)

## Template Documentation



# ExampleMoviesEntities Constructor

The default constructor for the Entity class

**Namespace:** [Template.Repository.Context](#)

**Assembly:** Template.Repository (in Template.Repository.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

C#

Copy

```
public ExampleMoviesEntities()
```

## ► See Also

Reference

[ExampleMoviesEntities Class](#)

[Template.Repository.Context Namespace](#)

## Template Documentation



# ExampleMoviesEntities Properties

The [ExampleMoviesEntities](#) type exposes the following members.

## Properties

Name	Description
 <a href="#">MOVIE_CATEGORY</a>	Represents the table MOVIE_CATEGORY of the database with the base CRUD functions
 <a href="#">MOVIES</a>	Represents the table MOVIES of the database with the base CRUD functions

[Top](#)

## See Also

[Reference](#)

[ExampleMoviesEntities Class](#)

[Template.Repository.Context Namespace](#)

## Template Documentation



# ExampleMoviesEntitiesMOVIE\_CAT Property

Represents the table MOVIE\_CATEGORY of the database with the base CRUD functions

**Namespace:** [Template.Repository.Context](#)

**Assembly:** Template.Repository (in Template.Repository.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

C#

[Copy](#)

```
public virtual DbSet<MOVIE_CATEGORY> MOVIE_CATEG
```

Property Value

Type: [DbSetMOVIE\\_CATEGORY](#)

## ► See Also

Reference

[ExampleMoviesEntities Class](#)

[Template.Repository.Context Namespace](#)

## Template Documentation



# ExampleMoviesEntitiesMOVIES Property

Represents the table MOVIES of the database with the base CRUD functions

**Namespace:** [Template.Repository.Context](#)

**Assembly:** Template.Repository (in Template.Repository.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

C#

Copy

```
public virtual DbSet<MOVIES> MOVIES { get; set; }
```

Property Value

Type: [DbSetMOVIES](#)

## ► See Also

Reference

[ExampleMoviesEntities Class](#)

[Template.Repository.Context Namespace](#)

## Template Documentation



# ExampleMoviesEntities Methods

The [ExampleMoviesEntities](#) type exposes the following members.

## ▪ Methods

Name	Description
 <a href="#">OnModelCreating</a>	Overrides the base function to only throw exception when is called, because this function is only called on a CodeFirst solution (Overrides <a href="#">DbContext.OnModelCreating(DbModelEventArgs)</a> )

[Top](#)

## ▪ See Also

Reference

[ExampleMoviesEntities Class](#)

[Template.Repository.Context Namespace](#)

## Template Documentation



# ExampleMoviesEntitiesOnModelCreate Method

Overrides the base function to only throw an exception when is called, because this function is only called on a CodeFirst solution

**Namespace:** [Template.Repository.Context](#)

**Assembly:** Template.Repository (in Template.Repository.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

C#

[Copy](#)

```
protected override void OnModelCreating(  
    DbModelBuilder modelBuilder  
)
```

### Parameters

*modelBuilder*

Type: [DbModelBuilder](#)

Used to a database schema

## ► Exceptions

Exception	Condition
<a href="#">UnintentionalCodeFirstException</a>	Throws when the function is called, because is a database first aproach

## ► See Also

## Reference

[ExampleMoviesEntities Class](#)

[Template.Repository.Context Namespace](#)

---

## Template Documentation



# MOVIE\_CATEGORY Class

Generated entity class by Entity Framework

## ► Inheritance Hierarchy

SystemObject  **BaseEntity**

Template.Repository.ContextMOVIE\_CATEGORY

**Namespace:** [Template.Repository.Context](#)

**Assembly:** Template.Repository (in Template.Repository.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

C#

Copy

```
public class MOVIE_CATEGORY : BaseEntity
```

The **MOVIE\_CATEGORY** type exposes the following members.

## ► Constructors

	Name	Description
≡	<a href="#">MOVIE_CATEGORY</a>	Generated default constructor by Entity Framework

[Top](#)

## ► Properties

	Name	Description
≡		

	<b>DESCRIPTION</b>	Generated primitive property by Entity Framework
	<b>ID</b>	Generated primitive property by Entity Framework
	<b>MOVIES</b>	Generated navigation property by Entity Framework

---

[Top](#)

## ▲ See Also

Reference

[Template.Repository.Context Namespace](#)

---

## Template Documentation



# MOVIE\_CATEGORY Constructor

Generated default constructor by Entity Framework

**Namespace:** [Template.Repository.Context](#)

**Assembly:** Template.Repository (in Template.Repository.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

C#

Copy

```
public MOVIE_CATEGORY()
```

## ► See Also

[Reference](#)

[MOVIE\\_CATEGORY Class](#)

[Template.Repository.Context Namespace](#)

## Template Documentation



# MOVIE\_CATEGORY Properties

The [MOVIE\\_CATEGORY](#) type exposes the following members.

## Properties

	Name	Description
	<a href="#">DESCRIPTION</a>	Generated primitive property by Entity Framework
	<a href="#">ID</a>	Generated primitive property by Entity Framework
	<a href="#">MOVIES</a>	Generated navigation property by Entity Framework

[Top](#)

## See Also

[Reference](#)

[MOVIE\\_CATEGORY Class](#)

[Template.Repository.Context Namespace](#)

## Template Documentation



# MOVIE\_CATEGORYDESCRIPTION Property

Generated primitive property by Entity Framework

**Namespace:** [Template.Repository.Context](#)

**Assembly:** Template.Repository (in Template.Repository.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

```
C#  
public string DESCRIPTION { get; set; }  
Copy
```

Property Value

Type: [String](#)

## ► See Also

Reference

[MOVIE\\_CATEGORY Class](#)

[Template.Repository.Context Namespace](#)

## Template Documentation



# MOVIE\_CATEGORYID Property

Generated primitive property by Entity Framework

**Namespace:** [Template.Repository.Context](#)

**Assembly:** Template.Repository (in Template.Repository.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

C#

Copy

```
public int ID { get; set; }
```

Property Value

Type: [Int32](#)

## ► See Also

[Reference](#)

[MOVIE\\_CATEGORY Class](#)

[Template.Repository.Context Namespace](#)

## Template Documentation



# MOVIE\_CATEGORYMOVIES Property

Generated navigation property by Entity Framework

**Namespace:** [Template.Repository.Context](#)

**Assembly:** Template.Repository (in Template.Repository.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

```
C# Copy
public virtual ICollection<MOVIES> MOVIES { get;
    <>1 <>2 <>3 <>4
```

Property Value

Type: [ICollectionMOVIES](#)

## ► See Also

Reference

[MOVIE\\_CATEGORY Class](#)

[Template.Repository.Context Namespace](#)

## Template Documentation



# MOVIES Class

Generated entity class by Entity Framework

## ► Inheritance Hierarchy

SystemObject  **BaseEntity**  
Template.Repository.ContextMOVIES

**Namespace:** [Template.Repository.Context](#)

**Assembly:** Template.Repository (in Template.Repository.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

C#

[Copy](#)

```
public class MOVIES : BaseEntity
```

The MOVIES type exposes the following members.

## ► Constructors

	Name	Description
≡	<a href="#">MOVIES</a>	Initializes a new instance of the MOVIES class

[Top](#)

## ► Properties

	Name	Description
	<a href="#">DESCRIPTION</a>	Generated primitive

		property by Entity Framework
	ID	Generated primitive property by Entity Framework
	ID_CATEGORY	Generated primitive property by Entity Framework
	IS_RENTED	Generated primitive property by Entity Framework
	MOVIE_CATEGORY	Generated navigation property by Entity Framework
	NAME	Generated primitive property by Entity Framework
	PRICE	Generated primitive property by Entity Framework

[Top](#)

## ▪ See Also

Reference

[Template.Repository.Context Namespace](#)

## Template Documentation



# MOVIES Constructor

Initializes a new instance of the [MOVIES](#) class

**Namespace:** [Template.Repository.Context](#)

**Assembly:** Template.Repository (in Template.Repository.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

C#

Copy

```
public MOVIES()
```

## ► See Also

[Reference](#)

[MOVIES Class](#)

[Template.Repository.Context Namespace](#)

## Template Documentation



# MOVIES Properties

The [MOVIES](#) type exposes the following members.

## Properties

	Name	Description
	<a href="#">DESCRIPTION</a>	Generated primitive property by Entity Framework
	<a href="#">ID</a>	Generated primitive property by Entity Framework
	<a href="#">ID_CATEGORY</a>	Generated primitive property by Entity Framework
	<a href="#">IS_RENTED</a>	Generated primitive property by Entity Framework
	<a href="#">MOVIE_CATEGORY</a>	Generated navigation property by Entity Framework
	<a href="#">NAME</a>	Generated primitive property by Entity Framework
	<a href="#">PRICE</a>	Generated primitive property by Entity Framework

[Top](#)

## ◀ See Also

[Reference](#)

[MOVIES Class](#)

[Template.Repository.Context Namespace](#)

---

## Template Documentation



# MOVIESDESCRIPTION Property

Generated primitive property by Entity Framework

**Namespace:** [Template.Repository.Context](#)

**Assembly:** Template.Repository (in Template.Repository.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

```
C# _____ Copy
public string DESCRIPTION { get; set; }
```

Property Value

Type: [String](#)

## ► See Also

Reference

[MOVIES Class](#)

[Template.Repository.Context Namespace](#)

## Template Documentation



# MOVIESID Property

Generated primitive property by Entity Framework

**Namespace:** [Template.Repository.Context](#)

**Assembly:** Template.Repository (in Template.Repository.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

C#

Copy

```
public int ID { get; set; }
```

Property Value

Type: [Int32](#)

## ► See Also

[Reference](#)

[MOVIES Class](#)

[Template.Repository.Context Namespace](#)

## Template Documentation



# MOVIESID\_CATEGORY Property

Generated primitive property by Entity Framework

**Namespace:** [Template.Repository.Context](#)

**Assembly:** Template.Repository (in Template.Repository.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

```
C# _____ Copy
public int ID_CATEGORY { get; set; }
```

Property Value

Type: [Int32](#)

## ► See Also

Reference

[MOVIES Class](#)

[Template.Repository.Context Namespace](#)

## Template Documentation



# MOVIESIS\_RENTED Property

Generated primitive property by Entity Framework

**Namespace:** [Template.Repository.Context](#)

**Assembly:** Template.Repository (in Template.Repository.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

C#

Copy

```
public bool IS_RENTED { get; set; }
```

Property Value

Type: [Boolean](#)

## ► See Also

[Reference](#)

[MOVIES Class](#)

[Template.Repository.Context Namespace](#)

## Template Documentation



# MOVIESMOVIE\_CATEGORY Property

Generated navigation property by Entity Framework

**Namespace:** [Template.Repository.Context](#)

**Assembly:** Template.Repository (in Template.Repository.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

```
C# Copy
public virtual MOVIE_CATEGORY MOVIE_CATEGORY { get; }
```

Property Value

Type: [MOVIE\\_CATEGORY](#)

## ► See Also

[Reference](#)

[MOVIES Class](#)

[Template.Repository.Context Namespace](#)

## Template Documentation



# MOVIESNAME Property

Generated primitive property by Entity Framework

**Namespace:** [Template.Repository.Context](#)

**Assembly:** Template.Repository (in Template.Repository.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

C#

Copy

```
public string NAME { get; set; }
```

Property Value

Type: [String](#)

## ► See Also

[Reference](#)

[MOVIES Class](#)

[Template.Repository.Context Namespace](#)

## Template Documentation



# MOVIESPRICE Property

Generated primitive property by Entity Framework

**Namespace:** [Template.Repository.Context](#)

**Assembly:** Template.Repository (in Template.Repository.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

C#

Copy

```
public decimal PRICE { get; set; }
```

Property Value

Type: [Decimal](#)

## ► See Also

[Reference](#)

[MOVIES Class](#)

[Template.Repository.Context Namespace](#)

## Template Documentation



# Template.Repository.Core Namespace

## ► Classes

	Class	Description
	<a href="#">GenericRepositoryTEntity</a>	Generic repository with all the basic functions of every repository, all repositories will inherit of this class
	<a href="#">UnitOfWork</a>	Works to encapsulate all repositories, all the repositories will work with one DbContext to encapsulate all the changes in one transaction

## Template Documentation



# GenericRepository Class

Generic repository with all the basic functions of every repository, all repositories will inherit of this class

## ► Inheritance Hierarchy

```
SystemObject Template.Repository.CoreGenericRepository
Template.Repository.ClassMoviesCategoryRepository
Template.Repository.ClassMoviesRepository
```

**Namespace:** [Template.Repository.Core](#)

**Assembly:** [Template.Repository](#) (in [Template.Repository.dll](#)) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

```
C# Copy
public class GenericRepository<TEntity> : IGeneri
where TEntity : BaseEntity
```

### Type Parameters

*TEntity*

The entity class than represents a database table

The [GenericRepository](#) type exposes the following members.

## ► Constructors

	Name	Description
≡	<a href="#">GenericRepository<tentity></tentity></a>	The GenericRepository

constructor

[Top](#)

## Methods

	Name	Description
	<a href="#">Add</a>	Add a new entity to the context, it will be added to the database on Save
	<a href="#">AddIEnumerable</a>	Add a IEnumerable of new entities to the context, they will be added to the database on Save
	<a href="#">Delete</a>	Marks the given entity as deleted, the entity will be deleted on Save
	<a href="#">DeleteBy</a>	Marks the entities that match with the given predicate as deleted, they will be deleted of the database on Save
	<a href="#">Edit</a>	Marks the passed entity as Modified, the changes will persist on Save
	<a href="#">FindBy</a>	Filters the results by the passed predicate
	<a href="#">Get</a>	Gets an unique TEntity from the database
	<a href="#">GetAll</a>	Get all the results of the

## IEntity

---

	<a href="#">Save</a>	Save all the changes made to the context
	<a href="#">SelectWhereTResult</a>	This function allows to select the specific properties and filter by where clause on database, not on memory, in that way it improves the performance

---

[Top](#)

## ▲ Fields

	Name	Description
	<a href="#">_DBEntities</a>	The DbContext of entity framework
	<a href="#">_Entity</a>	The DbSet that represents the database table to do all the queries

---

[Top](#)

## ▲ See Also

Reference  
[Template.Repository.Core Namespace](#)

## Template Documentation



# GenericRepository Constructor

The GenericRepository constructor

**Namespace:** [Template.Repository.Core](#)

**Assembly:** Template.Repository (in Template.Repository.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

C#

Copy

```
public GenericRepository(  
    DbContext entities  
)
```

## Parameters

*entities*

Type: [DbContext](#)

[Missing <param name="entities"/> documentation for

"M:Template.Repository.Core.GenericRepository`1.#ctor(System.Data.Entity.DbC

## ► See Also

[Reference](#)

[GenericRepository Class](#)

[Template.Repository.Core Namespace](#)

## Template Documentation



# GenericRepository Methods

The [GenericRepository](#) generic type exposes the following members.

## ▲ Methods

	Name	Description
≡	<a href="#">Add</a>	Add a new entity to the context, it will be added to the database on Save
≡	<a href="#">AddIEnumerable</a>	Add a IEnumerable of new entities to the context, they will be added to the database on Save
≡	<a href="#">Delete</a>	Marks the given entity as deleted, the entity will be deleted on Save
≡	<a href="#">DeleteBy</a>	Marks the entities that match with the given predicate as deleted, they will be deleted of the database on Save
≡	<a href="#">Edit</a>	Marks the passed entity as Modified, the changes will persist on Save
≡	<a href="#">FindBy</a>	Filters the results by the

		passed predicate
≡	<a href="#">Get</a>	Gets an unique TEntity from the database
≡	<a href="#">GetAll</a>	Get all the results of the IEntity
≡	<a href="#">Save</a>	Save all the changes made to the context
≡	<a href="#">SelectWhereTResult</a>	This function allows to select the specific properties and filter by where clause on database, not on memory, in that way it improves the performance

[Top](#)

## ▲ See Also

[Reference](#)

[GenericRepository Class](#)

[Template.Repository.Core Namespace](#)

## Template Documentation



# GenericRepositoryTEntityAdd Method

Add a new entity to the context, it will be added to the database on Save

**Namespace:** [Template.Repository.Core](#)

**Assembly:** Template.Repository (in Template.Repository.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

C#

[Copy](#)

```
public virtual TEntity Add(  
    TEntity entity  
)
```

### Parameters

*entity*

Type: [TEntity](#)

The entity to be added

### Return Value

Type: [TEntity](#)

The entity after being added

### Implements

[IGenericRepositoryTEntityAdd\(TEntity\)](#)

## ► See Also

[Reference](#)

[GenericRepositoryTEntity Class](#)

## Template.Repository.Core Namespace

---

## Template Documentation



# GenericRepositoryAddIEnum Method

Add a IEnumerable of new entities to the context, they will be added to the database on Save

**Namespace:** [Template.Repository.Core](#)

**Assembly:** Template.Repository (in Template.Repository.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

```
C# Copy
public virtual IEnumerable<TEntity> AddIEnumab]
    IEnumerable<TEntity> entities
)
```

### Parameters

*entities*

Type: [System.Collections.Generic.IEnumerable<TEntity>](#)

The list of entities to insert

### Return Value

Type: [IEnumerable<TEntity>](#)

The IEnumerable of entities after being added

### Implements

[IGenericRepository<TEntity>AddIEnumerable\(IEnumerable<TEntity>\)](#)

## ► See Also

[Reference](#)

GenericRepository Class  
Template.Repository.Core Namespace

---

## Template Documentation



# GenericRepository Delete Method

Marks the given entity as deleted, the entity will be deleted on Save

**Namespace:** [Template.Repository.Core](#)

**Assembly:** Template.Repository (in Template.Repository.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

C#

Copy

```
public virtual TEntity Delete(  
    TEntity entity  
)
```

### Parameters

*entity*

Type: [TEntity](#)

The entity to be removed

### Return Value

Type: [TEntity](#)

The deleted entity

### Implements

[IGenericRepositoryTEntityDelete\(TEntity\)](#)

## ► See Also

Reference

[GenericRepositoryTEntity Class](#)

[Template.Repository.Core Namespace](#)



## Template Documentation



# GenericRepositoryDeleteBy Method

Marks the entities that match with the given predicate as deleted, they will be deleted of the database on Save

**Namespace:** [Template.Repository.Core](#)

**Assembly:** Template.Repository (in Template.Repository.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

```
C# Copy
public int DeleteBy(
    Expression<Func< TEntity, bool>> predicate
)
```

## Parameters

*predicate*

Type: [System.Linq.Expressions.Expression<Func< TEntity, Boolean>>](#)  
The predicate to filter the result

## Return Value

Type: [Int32](#)

The number of affected rows entity

## Implements

[IGenericRepository< TEntity>DeleteBy\(Expression<Func< TEntity, Boolean>> predicate\)](#)

## ► See Also

## Reference

[GenericRepository Class](#)  
[Template.Repository.Core Namespace](#)

---

## Template Documentation



# GenericRepositoryEdit Method

Marks the passed entity as Modified, the changes will persist on Save

**Namespace:** [Template.Repository.Core](#)

**Assembly:** Template.Repository (in Template.Repository.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

```
C#  
public virtual TEntity Edit(  
    TEntity entity  
)
```

[Copy](#)

### Parameters

*entity*

Type: [TEntity](#)

[Missing <param name="entity"/> documentation for  
"M:Template.Repository.Core.GenericRepository`1.Edit(`0)"]

### Return Value

Type: [TEntity](#)

The modified entity

### Implements

[IGenericRepositoryTEntityEdit\(TEntity\)](#)

## ► See Also

### Reference

[GenericRepositoryTEntity Class](#)

## Template.Repository.Core Namespace

---

## Template Documentation



# GenericRepositoryFindBy Method

Filters the results by the passed predicate

**Namespace:** [Template.Repository.Core](#)

**Assembly:** Template.Repository (in Template.Repository.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

```
C# Copy
public virtual IEnumerable<TEntity> FindBy(
    Expression<Func<TEntity, bool>> predicate
)
```

### Parameters

*predicate*

Type: [System.Linq.Expressions.Expression<Func<TEntity, Boolean>>](#)  
The expression to filter the result

### Return Value

Type: [IEnumerable<TEntity>](#)

The IEnumerable with the results

### Implements

[IGenericRepository<TEntity>.FindBy\(Expression<Func<TEntity, Boolean>>\)](#)

## ► See Also

Reference

[GenericRepository<TEntity> Class](#)

## Template.Repository.Core Namespace

---

## Template Documentation



# GenericRepositoryGet Method

Gets an unique TEntity from the database

**Namespace:** [Template.Repository.Core](#)

**Assembly:** Template.Repository (in Template.Repository.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

```
C# Copy
public TEntity Get(
    Expression<Func<◀ ▶
```

### Parameters

*predicate*

Type: [System.Linq.Expressions.ExpressionFunc](#)  
The predicate to filter the result

### Return Value

Type: [TEntity](#)

The TEntity of the result

### Implements

[IGenericRepositoryGet\(ExpressionFunc\)](#)

## ► See Also

Reference

[GenericRepository Class](#)

## Template.Repository.Core Namespace

---

## Template Documentation



# GenericRepository GetAll Method

Get all the results of the IEntity

**Namespace:** [Template.Repository.Core](#)

**Assembly:** Template.Repository (in Template.Repository.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

```
C# Copy
public virtual IEnumerable<TEntity> GetAll()
```

Return Value

Type: [IEnumerable TEntity](#)

returns an IEnumerable with all the results

Implements

[IGenericRepositoryTEntityGetAll](#)

## ► See Also

Reference

[GenericRepositoryTEntity Class](#)

[Template.Repository.Core Namespace](#)

## Template Documentation



# GenericRepositorySave Method

Save all the changes made to the context

**Namespace:** [Template.Repository.Core](#)

**Assembly:** Template.Repository (in Template.Repository.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

C#

Copy

```
public virtual int Save()
```

Return Value

Type: [Int32](#)

The number of affected rows

Implements

[IGenericRepositoryTEntitySave](#)

## ► See Also

Reference

[GenericRepositoryTEntity Class](#)

[Template.Repository.Core Namespace](#)

## Template Documentation



# GenericRepositorySelectWhere Method

This function allows to select the specific properties and filter by where clause on database, not on memory, in that way it improves the performance

**Namespace:** [Template.Repository.Core](#)

**Assembly:** Template.Repository (in Template.Repository.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

```
C# Copy
public IEnumerable<TResult> SelectWhere<TResult>(
    Func<TEntity, TResult> selector,
    Expression<Func<TEntity, bool>> predicate
)
```

## Parameters

*selector*

Type: [System.Func<TEntity, TResult>](#)

The properties to select

*predicate*

Type: [System.Linq.Expressions.Expression<Func<TEntity, Boolean>>](#)

The where clause

## Type Parameters

*TResult*

The select result type

## Return Value

Type: [IEnumerable<TResult>](#)

The IEnumerable of the result

## Implements

[IGenericRepository< TEntity >.SelectWhere< TResult >\(Func< TEntity, TResult >, ExpressionFunc< TEntity >, Boolean\)](#)

## See Also

### Reference

[GenericRepository< TEntity > Class](#)

[Template.Repository.Core Namespace](#)

---

## Template Documentation



# GenericRepository Fields

The [GenericRepository](#) generic type exposes the following members.

## ► Fields

	Name	Description
	<a href="#">_DBEntities</a>	The DbContext of entity framework
	<a href="#">_Entity</a>	The DbSet that represents the database table to do all the queries

[Top](#)

## ► See Also

### Reference

[GenericRepository Class](#)

[Template.Repository.Core Namespace](#)

## Template Documentation



# GenericRepository\_DBEntities Field

The DbContext of entity framework

**Namespace:** [Template.Repository.Core](#)

**Assembly:** Template.Repository (in Template.Repository.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

```
C# _____ Copy
protected DbContext _DBEntities
```

Field Value

Type: [DbContext](#)

## ► See Also

Reference

[GenericRepository Class](#)

[Template.Repository.Core Namespace](#)

## Template Documentation



# GenericRepository\_Entity Field

The DbSet that represents the database table to do all the queries

**Namespace:** [Template.Repository.Core](#)

**Assembly:** Template.Repository (in Template.Repository.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

```
C# Copy
protected readonly IDbSet<TEntity> _Entity
```

Field Value

Type: [IDbSet](#)

## ► See Also

Reference

[GenericRepository Class](#)

[Template.Repository.Core Namespace](#)

## Template Documentation



# UnitOfWork Class

Works to encapsulate all repositories, all the repositories will work with one DBContext to encapsulate all the changes in one transaction

## ► Inheritance Hierarchy

[SystemObject](#) [Template.Repository.CoreUnitOfWork](#)

**Namespace:** [Template.Repository.Core](#)

**Assembly:** Template.Repository (in Template.Repository.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

```
C#  
public class UnitOfWork : IUnitOfWork, IDisposable
```

The [UnitOfWork](#) type exposes the following members.

## ► Constructors

	Name	Description
•	<a href="#">UnitOfWork</a>	Constructor of UnitOfWork

[Top](#)

## ► Properties

	Name	Description
•	<a href="#">Movies</a>	MoviesRepository



MoviesCategory

MoviesCategoryRepository

[Top](#)

## ► Methods

	Name	Description
	<a href="#">Commit</a>	Save changes across all the repositories in one transaction
	<a href="#">Dispose</a>	Dispose the DBContext of the UnitOfWork

[Top](#)

## ► Fields

	Name	Description
	<a href="#">_entities</a>	
	<a href="#">_Movies</a>	
	<a href="#">_MoviesCateogory</a>	

[Top](#)

## ► See Also

[Reference](#)

[Template.Repository.Core Namespace](#)

## Template Documentation



# UnitOfWork Constructor

Constructor of UnitOfWork

**Namespace:** [Template.Repository.Core](#)

**Assembly:** Template.Repository (in Template.Repository.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

C#

Copy

```
public UnitOfWork()
```

## ► See Also

Reference

[UnitOfWork Class](#)

[Template.Repository.Core Namespace](#)

## Template Documentation



# UnitOfWork Properties

The [UnitOfWork](#) type exposes the following members.

## Properties

	Name	Description
	<a href="#">Movies</a>	MoviesRepository
	<a href="#">MoviesCategory</a>	MoviesCategoryRepository

[Top](#)

## See Also

[Reference](#)

[UnitOfWork Class](#)

[Template.Repository.Core Namespace](#)

## Template Documentation



# UnitOfWorkMovies Property

MoviesRepository

**Namespace:** [Template.Repository.Core](#)

**Assembly:** Template.Repository (in Template.Repository.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

C#

Copy

```
public IMoviesRepository Movies { get; }
```

Property Value

Type: [IMoviesRepository](#)

Implements

[IUnitOfWorkMovies](#)

## ► See Also

Reference

[UnitOfWork Class](#)

[Template.Repository.Core Namespace](#)

## Template Documentation



# UnitOfWorkMoviesCategory Property

MoviesCategoryRepository

**Namespace:** [Template.Repository.Core](#)

**Assembly:** Template.Repository (in Template.Repository.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

```
C# Copy
public IMoviesCategoryRepository MoviesCategory {
```

Property Value

Type: [IMoviesCategoryRepository](#)

Implements

[IUnitOfWorkMoviesCategory](#)

## ► See Also

Reference

[UnitOfWork Class](#)

[Template.Repository.Core Namespace](#)

## Template Documentation



# UnitOfWork Methods

The [UnitOfWork](#) type exposes the following members.

## ▪ Methods

	Name	Description
≡	<a href="#">Commit</a>	Save changes across all the repositories in one transaction
≡	<a href="#">Dispose</a>	Dispose the DBContext of the UnitOfWork

[Top](#)

## ▪ See Also

Reference

[UnitOfWork Class](#)

[Template.Repository.Core Namespace](#)

## Template Documentation



# UnitOfWorkCommit Method

Save changes across all the repositories in one transaction

**Namespace:** [Template.Repository.Core](#)

**Assembly:** Template.Repository (in Template.Repository.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

C#

Copy

```
public int Commit()
```

Return Value

Type: [Int32](#)

The number of affected rows

Implements

[IUnitOfWorkCommit](#)

## ► See Also

Reference

[UnitOfWork Class](#)

[Template.Repository.Core Namespace](#)

## Template Documentation



# UnitOfWorkDispose Method

Dispose the DBContext of the UnitOfWork

**Namespace:** [Template.Repository.Core](#)

**Assembly:** Template.Repository (in Template.Repository.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

C#

Copy

```
public void Dispose()
```

Implements

[IDisposable.Dispose](#)

## ► See Also

[Reference](#)

[UnitOfWork Class](#)

[Template.Repository.Core Namespace](#)

## Template Documentation



# UnitOfWork Fields

## ► Fields

	Name	Description
	<a href="#">_entities</a>	
	<a href="#">_Movies</a>	
	<a href="#">_MoviesCategory</a>	

[Top](#)

## ► See Also

[Reference](#)

[UnitOfWork Class](#)

[Template.Repository.Core Namespace](#)

## Template Documentation



# UnitOfWork\_entities Field

[Missing <summary> documentation for  
"F:Template.Repository.Core.UnitOfWork.\_entities"]

**Namespace:** [Template.Repository.Core](#)

**Assembly:** Template.Repository (in Template.Repository.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

C#

Copy

```
private ExampleMoviesEntities _entities
```

Field Value

Type: [ExampleMoviesEntities](#)

## ► See Also

[Reference](#)

[UnitOfWork Class](#)

[Template.Repository.Core Namespace](#)

## Template Documentation



# UnitOfWork\_Movies Field

[Missing <summary> documentation for  
"F:Template.Repository.Core.UnitOfWork.\_Movies"]

**Namespace:** [Template.Repository.Core](#)

**Assembly:** Template.Repository (in Template.Repository.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

C#

Copy

```
private IMoviesRepository _Movies
```

Field Value

Type: [IMoviesRepository](#)

## ► See Also

[Reference](#)

[UnitOfWork Class](#)

[Template.Repository.Core Namespace](#)

## Template Documentation



# UnitOfWork\_MoviesCategory Field

[Missing <summary> documentation for  
"F:Template.Repository.Core.UnitOfWork.\_MoviesCategory"]

**Namespace:** [Template.Repository.Core](#)

**Assembly:** Template.Repository (in Template.Repository.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

```
C#  
private IMoviesCategoryRepository _MoviesCateogor  
[<>]
```

[Copy](#)

Field Value

Type: [IMoviesCategoryRepository](#)

## ► See Also

Reference

[UnitOfWork Class](#)

[Template.Repository.Core Namespace](#)

## Template Documentation



# Template.Repository.Core.Interface Namespace

## ▪ Interfaces

Interface	Description
<a href="#">IUnitOfWork</a>	Defines the UnitOfWork functions and repositories will work with

## Template Documentation



# IUnitOfWork Interface

Defines the UnitOfWork functions and repositories will work with

**Namespace:** [Template.Repository.Core.Interface](#)

**Assembly:** Template.Repository (in Template.Repository.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

C#

Copy

```
public interface IUnitOfWork : IDisposable
```

The [IUnitOfWork](#) type exposes the following members.

## ► Properties

	Name	Description
	<a href="#">Movies</a>	MoviesRepository
	<a href="#">MoviesCategory</a>	MoviesCategoryRepository

[Top](#)

## ► Methods

	Name	Description
	<a href="#">Commit</a>	Save changes across all the repositories in one transaction

[Top](#)

## ◀ See Also

Reference

[Template.Repository.Core.Interface Namespace](#)

---

## Template Documentation



# IUnitOfWork Properties

The [IUnitOfWork](#) type exposes the following members.

## ► Properties

	Name	Description
	<a href="#">Movies</a>	MoviesRepository
	<a href="#">MoviesCategory</a>	MoviesCategoryRepository

[Top](#)

## ► See Also

[Reference](#)

[IUnitOfWork Interface](#)

[Template.Repository.Core.Interface Namespace](#)

## Template Documentation



# IUnitOfWorkMovies Property

MoviesRepository

**Namespace:** [Template.Repository.Core.Interface](#)

**Assembly:** Template.Repository (in Template.Repository.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

C#

Copy

```
IMoviesRepository Movies { get; }
```

Property Value

Type: [IMoviesRepository](#)

## ► See Also

[Reference](#)

[IUnitOfWork Interface](#)

[Template.Repository.Core.Interface Namespace](#)

## Template Documentation



# IUnitOfWorkMoviesCategory Property

MoviesCategoryRepository

**Namespace:** [Template.Repository.Core.Interface](#)

**Assembly:** Template.Repository (in Template.Repository.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

```
C# Copy
IMoviesCategoryRepository MoviesCategory { get; }
```

Property Value

Type: [IMoviesCategoryRepository](#)

## ► See Also

Reference

[IUnitOfWork Interface](#)

[Template.Repository.Core.Interface Namespace](#)

## Template Documentation



# IUnitOfWork Methods

The [IUnitOfWork](#) type exposes the following members.

## ▪ Methods

	Name	Description
	<a href="#">Commit</a>	Save changes across all the repositories in one transaction

[Top](#)

## ▪ See Also

Reference

[IUnitOfWork Interface](#)

[Template.Repository.Core.Interface Namespace](#)

## Template Documentation



# IUnitOfWorkCommit Method

Save changes across all the repositories in one transaction

**Namespace:** [Template.Repository.Core.Interface](#)

**Assembly:** Template.Repository (in Template.Repository.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

C#

Copy

```
int Commit()
```

Return Value

Type: [Int32](#)

The number of affected rows

## ► See Also

Reference

[IUnitOfWork Interface](#)

[Template.Repository.Core.Interface Namespace](#)

## Template Documentation



# Template.Repository.Interface Namespace

## ▪ Interfaces

Interface	Description
<a href="#">IMoviesCategoryRepository</a>	Interface to define all the IMoviesCategoryRepository database actions
<a href="#">IMoviesRepository</a>	Interface to define all the IMoviesRepository database actions

## Template Documentation



# IMoviesCategoryRepository Interface

Interface to define all the IMoviesCategoryRepository database actions

**Namespace:** [Template.Repository.Interface](#)

**Assembly:** Template.Repository (in Template.Repository.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

```
C# Copy
public interface IMoviesCategoryRepository : IGer
```

The [IMoviesCategoryRepository](#) type exposes the following members.

## ► Methods

	Name	Description
	<a href="#">Add</a>	Add a new entity to the context, it will be added to the database on Save (Inherited from <a href="#">IGenericRepositoryTEntity</a> .)
	<a href="#">AddIEnumerable</a>	Add a IEnumerable of new entities to the context, they will be added to the database on Save (Inherited from <a href="#">IGenericRepositoryTEntity</a> .)

	<a href="#">Delete</a>	Marks the given entity as deleted, the entity will be deleted on Save (Inherited from <a href="#">IGenericRepositoryTEntity</a> .)
	<a href="#">DeleteBy</a>	Marks the entities that match with the given predicate as deleted, they will be deleted of the database on Save (Inherited from <a href="#">IGenericRepositoryTEntity</a> .)
	<a href="#">Edit</a>	Marks the passed entity as Modified, the changes will persist on Save (Inherited from <a href="#">IGenericRepositoryTEntity</a> .)
	<a href="#">FindBy</a>	Filters the results by the passed predicate (Inherited from <a href="#">IGenericRepositoryTEntity</a> .)
	<a href="#">Get</a>	Gets an unique TEntity from the database (Inherited from <a href="#">IGenericRepositoryTEntity</a> .)
	<a href="#">GetAll</a>	Get all the results of the IEntity (Inherited from <a href="#">IGenericRepositoryTEntity</a> .)
	<a href="#">Save</a>	Save all the changes made to the context (Inherited from

[IGenericRepository TEntity.\)](#)

---



#### [SelectWhereTResult](#)

This function allows to select the specific properties and filter by where clause on database, not on memory, in that way it improves the performace (Inherited from [IGenericRepository TEntity.](#))

---

[Top](#)

## ▲ See Also

[Reference](#)

[Template.Repository.Interface Namespace](#)

---

## Template Documentation



# IMoviesCategoryRepository Methods

The [IMoviesCategoryRepository](#) type exposes the following members.

## ▪ Methods

Name	Description
<a href="#">Add</a>	Add a new entity to the context, it will be added to the database on Save (Inherited from <a href="#">IGenericRepositoryTEntity</a> .)
<a href="#">AddIEnumerable</a>	Add a IEnumerable of new entities to the context, they will be added to the database on Save (Inherited from <a href="#">IGenericRepositoryTEntity</a> .)
<a href="#">Delete</a>	Marks the given entity as deleted, the entity will be deleted on Save (Inherited from <a href="#">IGenericRepositoryTEntity</a> .)
<a href="#">DeleteBy</a>	Marks the entities that match with the given predicate as deleted, they will be deleted of the database on Save (Inherited from

[IGenericRepositoryTEntity.\)](#)

 Edit	Marks the passed entity as Modified, the changes will persist on Save (Inherited from <a href="#">IGenericRepositoryTEntity</a> .)
 FindBy	Filters the results by the passed predicate (Inherited from <a href="#">IGenericRepositoryTEntity</a> .)
 Get	Gets an unique TEntity from the database (Inherited from <a href="#">IGenericRepositoryTEntity</a> .)
 GetAll	Get all the results of the IEntity (Inherited from <a href="#">IGenericRepositoryTEntity</a> .)
 Save	Save all the changes made to the context (Inherited from <a href="#">IGenericRepositoryTEntity</a> .)
 SelectWhereTResult	This function allows to select the specific properties and filter by where clause on database, not on memory, in that way it improves the performance (Inherited from <a href="#">IGenericRepositoryTEntity</a> .)

[Top](#)

## ▲ See Also

Reference

[IMoviesCategoryRepository Interface](#)

[Template.Repository.Interface Namespace](#)

---

## Template Documentation



# IMoviesRepository Interface

Interface to define all the IMoviesRepository database actions

**Namespace:** [Template.Repository.Interface](#)

**Assembly:** Template.Repository (in Template.Repository.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

```
C#  
public interface IMoviesRepository : IGenericRepo  
{  
    // Methods  
}
```

The [IMoviesRepository](#) type exposes the following members.

## ► Methods

	Name	Description
≡	<a href="#">Add</a>	Add a new entity to the context, it will be added to the database on Save (Inherited from <a href="#">IGenericRepositoryTEntity</a> .)
≡	<a href="#">AddIEnumerable</a>	Add a IEnumerable of new entities to the context, they will be added to the database on Save (Inherited from <a href="#">IGenericRepositoryTEntity</a> .)
≡	<a href="#">Delete</a>	Marks the given entity as deleted, the entity will be

deleted on Save  
(Inherited from  
[IGenericRepositoryTEntity](#).)

 <a href="#">DeleteBy</a>	Marks the entities that match with the given predicate as deleted, they will be deleted of the database on Save (Inherited from <a href="#">IGenericRepositoryTEntity</a> .)
 <a href="#">Edit</a>	Marks the passed entity as Modified, the changes will persist on Save (Inherited from <a href="#">IGenericRepositoryTEntity</a> .)
 <a href="#">FindBy</a>	Filters the results by the passed predicate (Inherited from <a href="#">IGenericRepositoryTEntity</a> .)
 <a href="#">Get</a>	Gets an unique TEntity from the database (Inherited from <a href="#">IGenericRepositoryTEntity</a> .)
 <a href="#">GetAll</a>	Get all the results of the IEntity (Inherited from <a href="#">IGenericRepositoryTEntity</a> .)
 <a href="#">Save</a>	Save all the changes made to the context (Inherited from <a href="#">IGenericRepositoryTEntity</a> .)



**SelectWhereTResult** This function allows to select the specific properties and filter by where clause on database, not on memory, in that way it improves the performance (Inherited from [IGenericRepository TEntity](#).)

---

[Top](#)

## ▲ See Also

Reference

[Template.Repository.Interface Namespace](#)

---

## Template Documentation



# IMoviesRepository Methods

The [IMoviesRepository](#) type exposes the following members.

## ▪ Methods

Name	Description
<a href="#">Add</a>	Add a new entity to the context, it will be added to the database on Save (Inherited from <a href="#">IGenericRepositoryTEntity</a> .)
<a href="#">AddIEnumerable</a>	Add a IEnumerable of new entities to the context, they will be added to the database on Save (Inherited from <a href="#">IGenericRepositoryTEntity</a> .)
<a href="#">Delete</a>	Marks the given entity as deleted, the entity will be deleted on Save (Inherited from <a href="#">IGenericRepositoryTEntity</a> .)
<a href="#">DeleteBy</a>	Marks the entities that match with the given predicate as deleted, they will be deleted of the database on Save (Inherited from <a href="#">IGenericRepositoryTEntity</a> .)

≡	Edit	Marks the passed entity as Modified, the changes will persist on Save (Inherited from <a href="#">IGenericRepositoryTEntity</a> .)
≡	FindBy	Filters the results by the passed predicate (Inherited from <a href="#">IGenericRepositoryTEntity</a> .)
≡	Get	Gets an unique TEntity from the database (Inherited from <a href="#">IGenericRepositoryTEntity</a> .)
≡	GetAll	Get all the results of the IEntity (Inherited from <a href="#">IGenericRepositoryTEntity</a> .)
≡	Save	Save all the changes made to the context (Inherited from <a href="#">IGenericRepositoryTEntity</a> .)
≡	SelectWhereTResult	This function allows to select the specific properties and filter by where clause on database, not on memory, in that way it improves the performance (Inherited from <a href="#">IGenericRepositoryTEntity</a> .)

[Top](#)

## ▲ See Also

Reference

[IMoviesRepository Interface](#)

[Template.Repository.Interface Namespace](#)

---

## Template Documentation



# Template.Service.Class Namespace

## ▪ Classes

Class	Description
 <a href="#">MoviesCategoryService</a>	Implements all the <a href="#">IMoviesCategoryService</a> functions with the base <a href="#">GenericServiceTModel</a> , <a href="#">TEntity</a> functions
 <a href="#">MoviesService</a>	Implements all the <a href="#">IMoviesService</a> functions with the base <a href="#">GenericServiceTModel</a> , <a href="#">TEntity</a> functions

## Template Documentation



# MoviesCategoryService Class

Implements all the [IMoviesCategoryService](#) functions with the base [GenericServiceTModel](#), [TEntity](#) functions

## ► Inheritance Hierarchy

```
SystemObject Template.Service.CoreGenericServiceMoviesCategory
MOVIE_CATEGORY
Template.Service.ClassMoviesCategoryService
```

**Namespace:** [Template.Service.Class](#)

**Assembly:** Template.Service (in Template.Service.dll) Version: 1.0.0.0  
(1.0.0.0)

## ► Syntax

```
C# Copy
public class MoviesCategoryService : GenericService<IMoviesCategoryService, IService<MoviesCa
```

The [MoviesCategoryService](#) type exposes the following members.

## ► Constructors

	Name	Description
≡	<a href="#">MoviesCategoryService</a>	Constructor of <a href="#">MoviesCategoryService</a> , Unity inject the <a href="#">IUnitOfWork</a> object for database transaction and pass the respective repository to the parent

for the most common functions

[Top](#)

## ◀ Methods

Name	Description
 <a href="#">Add</a>	Add an entity based on the passed model and insert it on database (Inherited from <a href="#">GenericServiceTModel</a> , <a href="#">TEntity</a> .)
 <a href="#">AddIEnumerableable</a>	Add an IEnumerable of entities based on the passed models IEnumerable and insert them on database (Inherited from <a href="#">GenericServiceTModel</a> , <a href="#">TEntity</a> .)
 <a href="#">Delete</a>	Add an entity based on the passed model and delete it on database (Inherited from <a href="#">GenericServiceTModel</a> , <a href="#">TEntity</a> .)
 <a href="#">DeleteBy</a>	Deletes any match with the given predicate (Inherited from <a href="#">GenericServiceTModel</a> , <a href="#">TEntity</a> .)

≡	Edit	Add an entity based on the passed model and delete it on database (Inherited from <a href="#">GenericServiceTModel</a> , <a href="#">TEntity</a> .)
≡	FindBy	Retuns an IEnumerable of models based on the expresion, retreives entities from the database and convert them to models (Inherited from <a href="#">GenericServiceTModel</a> , <a href="#">TEntity</a> .)
≡	Get	Get an unique model based on the expresion, retreives the entity and convert it to model (Inherited from <a href="#">GenericServiceTModel</a> , <a href="#">TEntity</a> .)
≡	GetAll	Get all the elements of the database, retreives entities from the database and convert them to models (Inherited from <a href="#">GenericServiceTModel</a> , <a href="#">TEntity</a> .)
≡	SelectWhereTResult	This function allows to select the specific properties and filter by

---

		where clause on database, not on memory, in that way it improves the performace (Inherited from <a href="#">GenericServiceTModel</a> , <a href="#">TEntity</a> .)
≡	<a href="#">ToEntity</a>	Converts a model to an entity (Inherited from <a href="#">GenericServiceTModel</a> , <a href="#">TEntity</a> .)
≡	<a href="#">ToEntityEnumerable</a>	Converts an IEnumerable of Models to an IEnumerable of entities (Inherited from <a href="#">GenericServiceTModel</a> , <a href="#">TEntity</a> .)
≡	<a href="#">ToEntityExpression</a>	Convert an expression of a model to an expression to an Entity, be careful to make an expression of with entities properties (Inherited from <a href="#">GenericServiceTModel</a> , <a href="#">TEntity</a> .)
≡	<a href="#">ToEntityFunctionTResult</a>	Convert a function of a model to a function of a TEntity, be careful to make an function with

---

		entities properties (Inherited from <a href="#">GenericServiceTModel</a> , <a href="#">TEntity</a> .)
	<a href="#">ToModel</a>	Converts an entity to a model (Inherited from <a href="#">GenericServiceTModel</a> , <a href="#">TEntity</a> .)
	<a href="#">ToModelEnumerable</a>	Convert (Inherited from <a href="#">GenericServiceTModel</a> , <a href="#">TEntity</a> .)

---

[Top](#)

## Fields

	Name	Description
	<a href="#">_mapper</a>	The IMapper object is used to map objects between TEntity and TModel types (Inherited from <a href="#">GenericServiceTModel</a> , <a href="#">TEntity</a> .)
	<a href="#">_repository</a>	The GenericRepository of TEntity to do all the common database functions (Inherited from <a href="#">GenericServiceTModel</a> , <a href="#">TEntity</a> .)
	<a href="#">_Uof</a>	

---

[Top](#)

## ◀ See Also

Reference

[Template.Service.Class Namespace](#)

---

## Template Documentation



# MoviesCategoryService Constructor

Constructor of [MoviesCategoryService](#), Unity inject the [IUnitOfWork](#) object for database transaction and pass the respective repository to the parent for the most common functions

**Namespace:** [Template.Service.Class](#)

**Assembly:** Template.Service (in Template.Service.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

C#

Copy

```
public MoviesCategoryService(  
    IUnitOfWork Uof  
)
```

## Parameters

*Uof*

Type: [Template.Repository.Core.InterfaceIUnitOfWork](#)  
The [IUnitOfWork](#) dependency

## ► See Also

### Reference

[MoviesCategoryService Class](#)  
[Template.Service.Class Namespace](#)

## Template Documentation



# MoviesCategoryService Methods

The [MoviesCategoryService](#) type exposes the following members.

## ▪ Methods

Name	Description
<a href="#">Add</a>	Add an entity based on the passed model and insert it on database (Inherited from <a href="#">GenericServiceTModel</a> , <a href="#">TEntity</a> .)
<a href="#">AddIEnumerableable</a>	Add an IEnumerable of entities based on the passed models IEnumerable and insert them on database (Inherited from <a href="#">GenericServiceTModel</a> , <a href="#">TEntity</a> .)
<a href="#">Delete</a>	Add an entity based on the passed model and delete it on database (Inherited from <a href="#">GenericServiceTModel</a> , <a href="#">TEntity</a> .)
<a href="#">DeleteBy</a>	Deletes any match with the given predicate (Inherited from <a href="#">GenericServiceTModel</a> ,

`TEntity.)`

---

≡  
Edit  
Add an entity based on the passed model and delete it on database  
(Inherited from `GenericServiceTModel`, `TEntity`.)

---

≡  
FindBy  
Retuns an `IEnumerable` of models based on the expresion, retreives entities from the database and convert them to models  
(Inherited from `GenericServiceTModel`, `TEntity`.)

---

≡  
Get  
Get an unique model based on the expresion, retreives the entity and convert it to model  
(Inherited from `GenericServiceTModel`, `TEntity`.)

---

≡  
GetAll  
Get all the elements of the database, retreives entities from the database and convert them to models  
(Inherited from `GenericServiceTModel`, `TEntity`.)

---

≡  
SelectWhereTResult  
This function allows to

select the specific properties and filter by where clause on database, not on memory, in that way it improves the performance  
(Inherited from [GenericServiceTModel](#), [TEntity](#).)

 <a href="#">ToEntity</a>	Converts a model to an entity (Inherited from <a href="#">GenericServiceTModel</a> , <a href="#">TEntity</a> .)
 <a href="#">ToEntityEnumerable</a>	Converts an IEnumerable of Models to an IEnumerable of entities (Inherited from <a href="#">GenericServiceTModel</a> , <a href="#">TEntity</a> .)
 <a href="#">ToEntityExpression</a>	Convert an expression of a model to an expression to an Entity, be careful to make an expression of with entities properties (Inherited from <a href="#">GenericServiceTModel</a> , <a href="#">TEntity</a> .)
 <a href="#">ToEntityFunctionTResult</a>	Convert a function of a model to a function of a TEntity, be careful to

make an function with entities properties  
(Inherited from [GenericServiceTModel](#), [TEntity](#).)

---

 <a href="#">ToModel</a>	Converts an entity to a model (Inherited from <a href="#">GenericServiceTModel</a> , <a href="#">TEntity</a> .)
 <a href="#">ToModelEnumerable</a>	Convert (Inherited from <a href="#">GenericServiceTModel</a> , <a href="#">TEntity</a> .)

---

[Top](#)

## ▲ See Also

Reference

[MoviesCategoryService Class](#)  
[Template.Service.Class Namespace](#)

## Template Documentation



# MoviesCategoryService Fields

## ► Fields

	Name	Description
	<a href="#">_mapper</a>	The IMapper object is used to map objects between TEntity and TModel types (Inherited from <a href="#">GenericServiceTModel</a> , <a href="#">TEntity</a> .)
	<a href="#">_repository</a>	The GenericRepository of TEntity to do all the common database functions (Inherited from <a href="#">GenericServiceTModel</a> , <a href="#">TEntity</a> .)
	<a href="#">_Uof</a>	

[Top](#)

## ► See Also

[Reference](#)

[MoviesCategoryService Class](#)

[Template.Service.Class Namespace](#)

## Template Documentation



# MoviesCategoryService\_Uof Field

[Missing <summary> documentation for  
"F:Template.Service.Class.MoviesCategoryService.\_Uof"]

**Namespace:** [Template.Service.Class](#)

**Assembly:** Template.Service (in Template.Service.dll) Version: 1.0.0.0  
(1.0.0.0)

## ► Syntax

C#

[Copy](#)

```
private readonly IUnitOfWork _Uof
```

Field Value

Type: [IUnitOfWork](#)

## ► See Also

[Reference](#)

[MoviesCategoryService Class](#)

[Template.Service.Class Namespace](#)

## Template Documentation



# MoviesService Class

Implements all the [IMoviesService](#) functions with the base [GenericService](#)[TModel](#), [TEntity](#) functions

## ► Inheritance Hierarchy

[SystemObject](#) [Template.Service.CoreGenericService](#)**MoviesModel**,  
**MOVIES**  
[Template.Service.ClassMoviesService](#)

**Namespace:** [Template.Service.Class](#)

**Assembly:** Template.Service (in Template.Service.dll) Version: 1.0.0.0  
(1.0.0.0)

## ► Syntax

```
C# Copy
public class MoviesService : GenericService<Movie>
    IMoviesService, IService<MoviesModel>
```

The [MoviesService](#) type exposes the following members.

## ► Constructors

	Name	Description
≡	<a href="#">MoviesService</a>	Constructor of <a href="#">MoviesService</a> , Unity inject the <a href="#">IUnitOfWork</a> object for database transaction and pass the respective repository to the parent for the most common functions

[Top](#)

## Methods

Name	Description
 <a href="#">Add</a>	Add an entity based on the passed model and insert it on database (Inherited from <a href="#">GenericServiceTModel</a> , <a href="#">TEntity</a> .)
 <a href="#">AddIEnumerableable</a>	Add an IEnumerable of entities based on the passed models IEnumerable and insert them on database (Inherited from <a href="#">GenericServiceTModel</a> , <a href="#">TEntity</a> .)
 <a href="#">Delete</a>	Add an entity based on the passed model and delete it on database (Inherited from <a href="#">GenericServiceTModel</a> , <a href="#">TEntity</a> .)
 <a href="#">DeleteBy</a>	Deletes any match with the given predicate (Inherited from <a href="#">GenericServiceTModel</a> , <a href="#">TEntity</a> .)
 <a href="#">Edit</a>	Add an entity based on the passed model and delete it on database

		(Inherited from <a href="#">GenericServiceTModel</a> , <a href="#">TEntity</a> .)
≡	<a href="#">FindBy</a>	Returns an IEnumerable of models based on the expression, retrieves entities from the database and convert them to models (Inherited from <a href="#">GenericServiceTModel</a> , <a href="#">TEntity</a> .)
≡	<a href="#">Get</a>	Get an unique model based on the expression, retrieves the entity and convert it to model (Inherited from <a href="#">GenericServiceTModel</a> , <a href="#">TEntity</a> .)
≡	<a href="#">GetAll</a>	Get all the elements of the database, retrieves entities from the database and convert them to models (Inherited from <a href="#">GenericServiceTModel</a> , <a href="#">TEntity</a> .)
≡	<a href="#">SelectWhereTResult</a>	This function allows to select the specific properties and filter by where clause on database, not on memory, in that way it

		improves the performace (Inherited from <a href="#">GenericServiceTModel</a> , <a href="#">TEntity</a> .)
≡	<a href="#">ToEntity</a>	Converts a model to an entity (Inherited from <a href="#">GenericServiceTModel</a> , <a href="#">TEntity</a> .)
≡	<a href="#">ToEntityEnumerable</a>	Converts an IEnumerable of Models to an IEnumerable of entities (Inherited from <a href="#">GenericServiceTModel</a> , <a href="#">TEntity</a> .)
≡	<a href="#">ToEntityExpression</a>	Convert an expression of a model to an expression to an Entity, be careful to make an expression of with entities properties (Inherited from <a href="#">GenericServiceTModel</a> , <a href="#">TEntity</a> .)
≡	<a href="#">ToEntityFunctionTResult</a>	Convert a function of a model to a function of a <a href="#">TEntity</a> , be careful to make an function with entities properties (Inherited from <a href="#">GenericServiceTModel</a> ,

[TEntity.\)](#)

 <a href="#">ToModel</a>	Converts an entity to a model (Inherited from <a href="#">GenericServiceTModel</a> , <a href="#">TEntity</a> .)
 <a href="#">ToModelEnumerable</a>	Convert (Inherited from <a href="#">GenericServiceTModel</a> , <a href="#">TEntity</a> .)

[Top](#)

## Fields

	Name	Description
 <a href="#">_mapper</a>	The IMapper object is used to map objects between TEntity and TModel types (Inherited from <a href="#">GenericServiceTModel</a> , <a href="#">TEntity</a> .)	
 <a href="#">_repository</a>	The GenericRepository of TEntity to do all the common database functions (Inherited from <a href="#">GenericServiceTModel</a> , <a href="#">TEntity</a> .)	
 <a href="#">_Uof</a>		

[Top](#)

## See Also

Reference

## Template.Service.Class Namespace

---

## Template Documentation



# MoviesService Constructor

Constructor of [MoviesService](#), Unity inject the [IUnitOfWork](#) object for database transaction and pass the respective repository to the parent for the most common functions

**Namespace:** [Template.Service.Class](#)

**Assembly:** Template.Service (in Template.Service.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

C#

Copy

```
public MoviesService(  
    IUnitOfWork Uof  
)
```

### Parameters

*Uof*

Type: [Template.Repository.Core.InterfaceIUnitOfWork](#)

The [IUnitOfWork](#) dependency

## ► See Also

Reference

[MoviesService Class](#)

[Template.Service.Class Namespace](#)

## Template Documentation



# MoviesService Methods

The [MoviesService](#) type exposes the following members.

## ▪ Methods

Name	Description
<a href="#">Add</a>	Add an entity based on the passed model and insert it on database (Inherited from <a href="#">GenericServiceTModel</a> , <a href="#">TEntity</a> .)
<a href="#">AddIEnumerableable</a>	Add an IEnumerable of entities based on the passed models IEnumerable and insert them on database (Inherited from <a href="#">GenericServiceTModel</a> , <a href="#">TEntity</a> .)
<a href="#">Delete</a>	Add an entity based on the passed model and delete it on database (Inherited from <a href="#">GenericServiceTModel</a> , <a href="#">TEntity</a> .)
<a href="#">DeleteBy</a>	Deletes any match with the given predicate (Inherited from <a href="#">GenericServiceTModel</a> ,

`TEntity.)`

---

≡  
Edit  
Add an entity based on the passed model and delete it on database  
(Inherited from `GenericServiceTModel`, `TEntity`.)

---

≡  
FindBy  
Retuns an `IEnumerable` of models based on the expresion, retreives entities from the database and convert them to models  
(Inherited from `GenericServiceTModel`, `TEntity`.)

---

≡  
Get  
Get an unique model based on the expresion, retreives the entity and convert it to model  
(Inherited from `GenericServiceTModel`, `TEntity`.)

---

≡  
GetAll  
Get all the elements of the database, retreives entities from the database and convert them to models  
(Inherited from `GenericServiceTModel`, `TEntity`.)

---

≡  
SelectWhereTResult  
This function allows to

select the specific properties and filter by where clause on database, not on memory, in that way it improves the performance  
(Inherited from [GenericServiceTModel](#), [TEntity](#).)

 <a href="#">ToEntity</a>	Converts a model to an entity (Inherited from <a href="#">GenericServiceTModel</a> , <a href="#">TEntity</a> .)
 <a href="#">ToEntityEnumerable</a>	Converts an IEnumerable of Models to an IEnumerable of entities (Inherited from <a href="#">GenericServiceTModel</a> , <a href="#">TEntity</a> .)
 <a href="#">ToEntityExpression</a>	Convert an expression of a model to an expression to an Entity, be careful to make an expression of with entities properties (Inherited from <a href="#">GenericServiceTModel</a> , <a href="#">TEntity</a> .)
 <a href="#">ToEntityFunctionTResult</a>	Convert a function of a model to a function of a TEntity, be careful to

make an function with entities properties  
(Inherited from [GenericServiceTModel](#), [TEntity](#).)

---

 <a href="#">ToModel</a>	Converts an entity to a model (Inherited from <a href="#">GenericServiceTModel</a> , <a href="#">TEntity</a> .)
 <a href="#">ToModelEnumerable</a>	Convert (Inherited from <a href="#">GenericServiceTModel</a> , <a href="#">TEntity</a> .)

---

[Top](#)

## ▲ See Also

[Reference](#)

[MoviesService Class](#)

[Template.Service.Class Namespace](#)

## Template Documentation



# MoviesService Fields

## ► Fields

	Name	Description
	<a href="#">_mapper</a>	The IMapper object is used to map objects between TEntity and TModel types (Inherited from <a href="#">GenericServiceTModel</a> , <a href="#"><tentity>.</tentity></a> )
	<a href="#">_repository</a>	The GenericRepository of TEntity to do all the common database functions (Inherited from <a href="#">GenericServiceTModel</a> , <a href="#"> TEntity.</a> )
	<a href="#">_Uof</a>	

[Top](#)

## ► See Also

[Reference](#)

[MoviesService Class](#)

[Template.Service.Class Namespace](#)

## Template Documentation



# MoviesService\_Uof Field

[Missing <summary> documentation for  
"F:Template.Service.Class.MoviesService.\_Uof"]

**Namespace:** Template.Service.Class

**Assembly:** Template.Service (in Template.Service.dll) Version: 1.0.0.0  
(1.0.0.0)

## ► Syntax

C#

Copy

```
private readonly IUnitOfWork _Uof
```

Field Value

Type: [IUnitOfWork](#)

## ► See Also

[Reference](#)

[MoviesService Class](#)

[Template.Service.Class Namespace](#)

## Template Documentation



# Template.Service.Core Namespace

## ► Classes

Class	Description
 <a href="#">GenericServiceTModel, TEntity</a>	Works as a Generic Service, to use the most commons functions of every service and encapsule them here, to avoid duplicated code. This class implements <a href="#">IServiceTModel</a> with the common functions and <a href="#">IAutoMapperTModel, TEntity</a> , to map objects between DTO and Entities and viceverse, The navigation properties will be mapping on a deep of 5 objects by default, if you wanna change that, change de value of MaxDeep on the <a href="#">GenericServiceTModel, TEntity</a> constructor

## Template Documentation



# GenericServiceTModel, TEntity Class

Works as a Generic Service, to use the most commons functions of every service and encapsule them here, to avoid duplicated code. This class implements [IServiceTModel](#) with the common functions and [IAutoMapperTModel, TEntity](#), to map objects between DTO and Entities and viceverse, The navigation properties will be mapping on a deep of 5 objects by default, if you wanna change that, change the value of MaxDeep on the [GenericServiceTModel, TEntity](#) constructor

## ► Inheritance Hierarchy

[SystemObject](#) [Template.Service.CoreGenericServiceTModel, TEntity](#)  
[Template.Service.ClassMoviesCategoryService](#)  
[Template.Service.ClassMoviesService](#)

**Namespace:** [Template.Service.Core](#)

**Assembly:** Template.Service (in Template.Service.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

C#

Copy

```
public abstract class GenericService<TModel, TEntity>
    : IAutoMapper<TModel, TEntity>
where TModel : BaseModel
where TEntity : BaseEntity
```

Type Parameters

*TModel*

The model class of the service

*TEntity*

The entity class of the service

The [GenericServiceTModel](#), [TEntity](#) type exposes the following members.

## ► Constructors

Name	Description
 <a href="#">GenericServiceTModel</a> , <a href="#">TEntity</a>	Add an instance of GenericService, also configures the mapping

[Top](#)

## ► Methods

Name	Description
 <a href="#">Add</a>	Add an entity based on the passed model and insert it on database
 <a href="#">AddIEnumerableable</a>	Add an IEnumerable of entities based on the passed models IEnumerable and insert them on database
 <a href="#">Delete</a>	Add an entity based on the passed model and delete it on database
 <a href="#">DeleteBy</a>	Deletes any match with the given predicate

≡♡	<a href="#">Edit</a>	Add an entity based on the passed model and delete it on database
≡♡	<a href="#">FindBy</a>	Retuns an IEnumerable of models based on the expresion, retreives entities from the database and convert them to models
≡♡	<a href="#">Get</a>	Get an unique model based on the expresion, retreives the entity and convert it to model
≡♡	<a href="#">GetAll</a>	Get all the elements of the database, retreives entities from the database and convert them to models
≡♡	<a href="#">SelectWhereTResult</a>	This function allows to select the specific properties and filter by where clause on database, not on memory, in that way it improves the performace
≡♡	<a href="#">ToEntity</a>	Converts a model to an entity
≡♡	<a href="#">ToEntityEnumerable</a>	Converts an IEnumerable of Models to an IEnumerable of

entities

≡	<a href="#">ToEntityExpression</a>	Convert an expression of a model to an expression to an Entity, be careful to make an expression of with entities properties
≡	<a href="#">ToEntityFunctionTResult</a>	Convert a function of a model to a function of a TEntity, be careful to make an function with entities properties
≡	<a href="#">ToModel</a>	Converts an entity to a model
≡	<a href="#">ToModelEnumerable</a>	Convert

[Top](#)

## Fields

	Name	Description
≡	<a href="#">_mapper</a>	The IMapper object is used to map objects between TEntity and TModel types
≡	<a href="#">_repository</a>	The GenericRepository of TEntity to do all the common database functions

[Top](#)

## See Also

## Reference

### [Template.Service.Core Namespace](#)

---

## Template Documentation



# GenericService{TModel, TEntity} Constructor

Add an instance of GenericService, also configures the mapping

**Namespace:** [Template.Service.Core](#)

**Assembly:** Template.Service (in Template.Service.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

```
C#  
public GenericService(  
    IGenericRepository< repository  
)
```

[Copy](#)

### Parameters

*repository*

Type: [Template.Core.InterfaceIGenericRepository{TEntity}](#)

The repository of the Service

## ► See Also

### Reference

[GenericService{TModel, TEntity} Class](#)

[Template.Service.Core Namespace](#)

## Template Documentation



# GenericServiceTModel, TEntity Methods

The [GenericServiceTModel, TEntity](#) generic type exposes the following members.

## Methods

Name	Description
<a href="#">Add</a>	Add an entity based on the passed model and insert it on database
<a href="#">AddIEnumerable</a>	Add an IEnumerable of entities based on the passed models IEnumerable and insert them on database
<a href="#">Delete</a>	Add an entity based on the passed model and delete it on database
<a href="#">DeleteBy</a>	Deletes any match with the given predicate
<a href="#">Edit</a>	Add an entity based on the passed model and delete it on database
<a href="#">FindBy</a>	Retuns an IEnumerable of models based on the expresion, retreives

		entities from the database and convert them to models
≡♡	<a href="#">Get</a>	Get an unique model based on the expresion, retreives the entity and convert it to model
≡♡	<a href="#">GetAll</a>	Get all the elements of the database, retreives entities from the database and convert them to models
≡♡	<a href="#">SelectWhereTResult</a>	This function allows to select the specific properties and filter by where clause on database, not on memory, in that way it improves the performace
≡♡	<a href="#">ToEntity</a>	Converts a model to an entity
≡♡	<a href="#">ToEntityEnumerable</a>	Converts an IEnumerable of Models to an IEnumerable of entities
≡♡	<a href="#">ToEntityExpression</a>	Convert an expression of a model to an expression to an Entity, be careful to make an

		expression of with entities properties
≡	<a href="#">ToEntityFunctionTResult</a>	Convert a function of a model to a function of a TEntity, be careful to make an function with entities properties
≡	<a href="#">ToModel</a>	Converts an entity to a model
≡	<a href="#">ToModelEnumerable</a>	Convert

[Top](#)

## ▲ See Also

### Reference

[GenericServiceTModel, TEntity Class](#)  
[Template.Service.Core Namespace](#)

## Template Documentation



# GenericService{TModel}, TEntityAdd Method

Add an entity based on the passed model and insert it on database

**Namespace:** [Template.Service.Core](#)

**Assembly:** Template.Service (in Template.Service.dll) Version: 1.0.0.0  
(1.0.0.0)

## ► Syntax

```
C# Copy
public virtual void Add(
    TModel model
)
```

### Parameters

*model*

Type: [TModel](#)

The model to convert in a entity

Implements

[IServiceTModelAdd\(TModel\)](#)

## ► Exceptions

Exception	Condition
<a href="#">ArgumentNullException</a>	Throws when model parameter is null

## ► See Also

## Reference

[GenericServiceTModel, TEntity Class](#)  
[Template.Service.Core Namespace](#)

---

## Template Documentation



# GenericServiceTModel, TEntityAddIEnumetable Method

Add an IEnumerable of entities based on the passed models  
IEnumerable amd insert them on database

**Namespace:** [Template.Service.Core](#)

**Assembly:** Template.Service (in Template.Service.dll) Version: 1.0.0.0  
(1.0.0.0)

## ► Syntax

C#

Copy

```
public void AddIEnumetable(  
    IEnumerable<TModel> models  
)
```

### Parameters

*models*

Type: [System.Collections.GenericIEnumableTModel](#)

THe model IEnumerable to convert an entity IEnumerable

### Implements

[IServiceTModelAddIEnumetable\(IEnumerableTModel\)](#)

## ► See Also

Reference

[GenericServiceTModel, TEntity Class](#)

[Template.Service.Core Namespace](#)

## Template Documentation



# GenericService{TModel}, TEntityDelete Method

Add an entity based on the passed model and delete it on database

**Namespace:** [Template.Service.Core](#)

**Assembly:** Template.Service (in Template.Service.dll) Version: 1.0.0.0  
(1.0.0.0)

## ► Syntax

```
C# Copy
public virtual void Delete(
    TModel model
)
```

### Parameters

*model*

Type: [TModel](#)

The model to convert in a entity

Implements

[IServiceTModelDelete\(TModel\)](#)

## ► Exceptions

Exception	Condition
<a href="#">ArgumentNullException</a>	Throws when model parameter is null

## ► See Also

## Reference

[GenericServiceTModel, TEntity Class](#)  
[Template.Service.Core Namespace](#)

---

## Template Documentation



# GenericServiceTModel, TEntityDeleteBy Method

Deletes any match with the given predicate

**Namespace:** [Template.Service.Core](#)

**Assembly:** Template.Service (in Template.Service.dll) Version: 1.0.0.0  
(1.0.0.0)

## ► Syntax

```
C# Copy
public int DeleteBy(
    Expression<Func<TModel, bool>> predicate
)
```

### Parameters

*predicate*

Type: [System.Linq.Expressions.Expression<Func<TModel, Boolean>>](#)

The predicate to filter the result

### Return Value

Type: [Int32](#)

The number of affected rows entity

### Implements

[IServiceTModelDeleteBy\(ExpressionFunc<TModel, Boolean>\)](#)

## ► See Also

### Reference

[GenericServiceTModel, TEntity Class](#)

## Template.Service.Core Namespace

---

## Template Documentation



# GenericServiceTModel, TEntityEdit Method

Add an entity based on the passed model and delete it on database

**Namespace:** [Template.Service.Core](#)

**Assembly:** Template.Service (in Template.Service.dll) Version: 1.0.0.0  
(1.0.0.0)

## ► Syntax

```
C#  
public virtual void Edit(  
    TModel model  
)
```

[Copy](#)

### Parameters

*model*

Type: [TModel](#)

### Implements

[IServiceTModelEdit\(TModel\)](#)

## ► See Also

### Reference

[GenericServiceTModel, TEntity Class](#)

[Template.Service.Core Namespace](#)

## Template Documentation



# GenericService{TModel}, TEntityFindBy Method

Returns an IEnumerable of models based on the expression, retrieves entities from the database and convert them to models

**Namespace:** [Template.Service.Core](#)

**Assembly:** Template.Service (in Template.Service.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

C#

Copy

```
public virtual IEnumerable<TModel> FindBy(  
    Expression<Func<TModel, bool>> predicate  
)
```

### Parameters

*predicate*

Type: [System.Linq.Expressions.Expression<Func<TModel, Boolean>>](#)

The expression to filter the selection

### Return Value

Type: [IEnumerable<TModel>](#)

The IEnumerable of models based on the predicate

### Implements

[IServiceTModelFindBy\(Expression<Func<TModel, Boolean>>\)](#)

## ► See Also

### Reference

GenericServiceTModel, TEntity Class  
Template.Service.Core Namespace

---

## Template Documentation



# GenericService{TModel}, TEntityGet Method

Get an unique model based on the expression, retrieves the entity and convert it to model

**Namespace:** [Template.Service.Core](#)

**Assembly:** Template.Service (in Template.Service.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

C#

[Copy](#)

```
public virtual TModel Get(  
    Expression<Func<TModel, bool>> predicate  
)
```

### Parameters

*predicate*

Type: [System.Linq.Expressions.ExpressionFunc{TModel, Boolean}](#)

The expression to filter the selection

### Return Value

Type: [TModel](#)

The model based on the predicate

### Implements

[IServiceTModelGet\(ExpressionFunc{TModel, Boolean}\)](#)

## ► See Also

### Reference

GenericServiceTModel, TEntity Class  
Template.Service.Core Namespace

---

## Template Documentation



# GenericServiceTModel, TEntity GetAll Method

Get all the elements of the database, retrieves entities from the database and convert them to models

**Namespace:** [Template.Service.Core](#)

**Assembly:** Template.Service (in Template.Service.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

C#

[Copy](#)

```
public virtual IEnumerable<TModel> GetAll()
```

Return Value

Type: [IEnumerable TModel](#)

The IEnumerable of models

Implements

[IServiceTModelGetAll](#)

## ► See Also

Reference

[GenericServiceTModel, TEntity Class](#)

[Template.Service.Core Namespace](#)

## Template Documentation



# GenericService<TModel>, TEntitySelectWhere<TResult> Method

This function allows to select the specific properties and filter by where clause on database, not on memory, in that way it improves the performance

**Namespace:** [Template.Service.Core](#)

**Assembly:** Template.Service (in Template.Service.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

C#

[Copy](#)

```
public IEnumerable<TResult> SelectWhere<TResult>(  
    Func<TModel, TResult> selector,  
    Expression<Func<TModel, bool>> predicate  
)
```

## Parameters

*selector*

Type: [System.Func<TModel, TResult>](#)

The properties to select

*predicate*

Type: [System.Linq.Expressions.Expression<Func<TModel, Boolean>>](#)

The where clause

## Type Parameters

*TResult*

The select result type

## Return Value

Type: [IEnumerable{TResult}](#)

The IEnumerable of the result

## Implements

[IServiceTModelSelectWhereTResult\(Func{TModel, TResult}, ExpressionFunc{TModel, Boolean}\)](#)

## See Also

Reference

[GenericServiceTModel, TEntity Class](#)  
[Template.Service.Core Namespace](#)

---

## Template Documentation



# GenericServiceTModel, TEntityToEntity Method

Converts a model to an entity

**Namespace:** [Template.Service.Core](#)

**Assembly:** Template.Service (in Template.Service.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

```
C#  
public TEntity ToEntity(  
    TModel model  
)
```

[Copy](#)

### Parameters

*model*

Type: [TModel](#)

The model to be converted

### Return Value

Type: [TEntity](#)

The entity object

### Implements

[IAutoMapperTModel](#), [TEntityToEntity\(TModel\)](#)

## ► See Also

### Reference

[GenericServiceTModel, TEntity Class](#)

[Template.Service.Core Namespace](#)



## Template Documentation



# GenericService{TModel}, TEntityToEntityEnumerable Method

Converts an IEnumerable of Models to an IEnumerable of entities

**Namespace:** [Template.Service.Core](#)

**Assembly:** Template.Service (in Template.Service.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

```
C# Copy
public virtual IEnumerable<TEntity> ToEntityEnumerable<TModel> enumModel
)
```

### Parameters

*enumModel*

Type: [System.Collections.Generic.IEnumerable{TModel}](#)

The IEnumerable of Models to be converted

### Return Value

Type: [IEnumerable{TEntity}](#)

The IEnumerable of entities

### Implements

[IAutoMapperTModel](#),

[TEntityToEntityEnumerable\(IEnumerable{TModel}\)](#)

## ► See Also

## Reference

[GenericServiceTModel, TEntity Class](#)  
[Template.Service.Core Namespace](#)

---

## Template Documentation



# GenericService{TModel}, TEntityToEntityExpression Method

Convert an expression of a model to an expression to an Entity, be careful to make an expression of with entities properties

**Namespace:** [Template.Service.Core](#)

**Assembly:** Template.Service (in Template.Service.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

```
C# Copy
public Expression<Func< TEntity, bool>> ToEntityE>
    Expression<Func< TModel, bool>> predicate
)
```

### Parameters

*predicate*

Type: [System.Linq.Expressions.ExpressionFunc{TModel, Boolean}](#)

The expression to convert

### Return Value

Type: [ExpressionFunc{TEntity, Boolean}](#)

The TEntity expression

### Implements

[IAutoMapperTModel](#),

[TEntityToEntityExpression\(ExpressionFunc{TModel, Boolean}\)](#)

## ▲ See Also

### Reference

[GenericServiceTModel](#),  [TEntity Class](#)  
[Template.Service.Core Namespace](#)

---

## Template Documentation



# GenericService<TModel>, TEntityToEntityFunction<TResult> Method

Convert a function of a model to a function of a TEntity, be careful to make an function with entities properties

**Namespace:** [Template.Service.Core](#)

**Assembly:** Template.Service (in Template.Service.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

```
C# Copy
public Func<
```

### Parameters

*selector*

Type: [SystemFunc<TModel, TResult>](#)

The selector to convert

### Type Parameters

*TResult*

The result type

### Return Value

Type: [Func< TEntity, TResult >](#)

The TEntity selector

Implements

[IAutoMapperTModel](#), [TEntityToEntityFunctionTResult\(FuncTModel, TResult\)](#)

## See Also

Reference

[GenericServiceTModel](#), [TEntity Class](#)

[Template.Service.Core Namespace](#)

---

## Template Documentation



# GenericServiceTModel, TEntityToModel Method

Converts an entity to a model

**Namespace:** [Template.Service.Core](#)

**Assembly:** Template.Service (in Template.Service.dll) Version: 1.0.0.0  
(1.0.0.0)

## ► Syntax

C#

Copy

```
public virtual TModel ToModel(  
    TEntity entity  
)
```

### Parameters

*entity*

Type: [TEntity](#)

The entity to be converted

### Return Value

Type: [TModel](#)

The model object

### Implements

[IAutoMapperTModel](#), [TEntityToModel\(TEntity\)](#)

## ► See Also

### Reference

[GenericServiceTModel, TEntity Class](#)

[Template.Service.Core Namespace](#)



## Template Documentation



# GenericService<TModel> TEntityToModelEnumerable Method

Convert

**Namespace:** [Template.Service.Core](#)

**Assembly:** Template.Service (in Template.Service.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

```
C# Copy
public virtual IEnumerable<TModel> ToModelEnumerable(
    IEnumerable<TEntity> enumEntity
)
```

## Parameters

*enumEntity*

Type: [System.Collections.Generic.IEnumerable<TEntity>](#)

## Return Value

Type: [IEnumerable<TModel>](#)

The IEnumerable of models

## Implements

[IAutoMapperTModel](#),

[TEntityToModelEnumerable\(IEnumerable<TEntity>\)](#)

## ► See Also

## Reference

[GenericServiceTModel, TEntity Class](#)  
[Template.Service.Core Namespace](#)

---

## Template Documentation



# GenericServiceTModel, TEntity Fields

## Fields

	Name	Description
	<a href="#">_mapper</a>	The IMapper object is used to map objects between TEntity and TModel types
	<a href="#">_repository</a>	The GenericRepository of TEntity to do all the common database functions

[Top](#)

## See Also

### Reference

[GenericServiceTModel, TEntity Class](#)  
[Template.Service.Core Namespace](#)

## Template Documentation



# GenericServiceTModel, TEntity\_mapper Field

The IMapper object is used to map objects between TEntity and TModel types

**Namespace:** [Template.Service.Core](#)

**Assembly:** Template.Service (in Template.Service.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

C#

[Copy](#)

```
private readonly IMapper _mapper
```

Field Value

Type: **IMapper**

## ► See Also

[Reference](#)

[GenericServiceTModel, TEntity Class](#)

[Template.Service.Core Namespace](#)

## Template Documentation



# GenericServiceTModel, TEntity\_repository Field

The GenericRepository of TEntity to do all the common database functions

**Namespace:** [Template.Service.Core](#)

**Assembly:** Template.Service (in Template.Service.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

C#

[Copy](#)

```
private readonly IGenericRepository<TEntity> _rep
```

Field Value

Type: [IGenericRepository<TEntity>](#)

## ► See Also

Reference

[GenericServiceTModel, TEntity Class](#)

[Template.Service.Core Namespace](#)

## Template Documentation



# Template.Service.Core.Interface Namespace

## ▪ Interfaces

	Interface	Description
	<a href="#">IAutoMapperTModel</a> , <a href="#">TEntity</a>	Defines the basic automapper methods
	<a href="#">IServiceTModel</a>	Defines all the Generic Service functions

## Template Documentation



# IAutoMapper<TModel, TEntity> Interface

Defines the basic automapper methods

**Namespace:** [Template.Service.Core.Interface](#)

**Assembly:** Template.Service (in Template.Service.dll) Version: 1.0.0.0  
(1.0.0.0)

## ► Syntax

C#

[Copy](#)

```
internal interface IAutoMapper<TModel, TEntity>
where TModel : BaseModel
where TEntity : BaseEntity
```

## Type Parameters

*TModel*

The TModel DTO type

*TEntity*

The Entity Framework entity type

The [IAutoMapper<TModel, TEntity>](#) type exposes the following members.

## ► Methods

	Name	Description
≡	<a href="#">ToEntity</a>	Converts a model to an entity
≡	<a href="#">ToEntityEnumerable</a>	Converts an

IEnumerable of Models  
to an IEnumerable of  
entities

---

 <a href="#">ToEntityExpression</a>	Convert an expression of a model to an expression to an Entity, be careful to make an expression of with entities properties
 <a href="#">ToEntityFunctionTResult</a>	Convert a function of a model to a function of a TEntity, be careful to make an function with entities properties
 <a href="#">ToModel</a>	Converts an entity to a model
 <a href="#">ToModelEnumerable</a>	Convert an IEnumerable of entities to an IEnumerable of models

---

[Top](#)

## ▲ See Also

[Reference](#)

[Template.Service.Core.Interface Namespace](#)

## Template Documentation



# IAutoMapper<TModel, TEntity> Methods

The [IAutoMapper<TModel, TEntity>](#) generic type exposes the following members.

## ▪ Methods

	Name	Description
≡	<a href="#">ToEntity</a>	Converts a model to an entity
≡	<a href="#">ToEntityEnumerable</a>	Converts an IEnumerable of Models to an IEnumerable of entities
≡	<a href="#">ToEntityExpression</a>	Convert an expression of a model to an expression to an Entity, be careful to make an expression of with entities properties
≡	<a href="#">ToEntityFunctionTResult</a>	Convert a function of a model to a function of a TEntity, be careful to make an function with entities properties
≡	<a href="#">ToModel</a>	Converts an entity to a model



## ToModelEnumerable

Convert an  
IEnumerable of entities  
to an IEnumerable of  
models

---

[Top](#)

## See Also

Reference

[IAutoMapperTModel, TEntity Interface](#)

[Template.Service.Core.Interface Namespace](#)

---

## Template Documentation



# IAutoMapperTModel, TEntityToEntity Method

Converts a model to an entity

**Namespace:** [Template.Service.Core.Interface](#)

**Assembly:** Template.Service (in Template.Service.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

```
C#  
TEntity ToEntity(  
    TModel model  
)
```

[Copy](#)

### Parameters

*model*

Type: [TModel](#)

The model to be converted

### Return Value

Type: [TEntity](#)

The entity object

## ► See Also

[Reference](#)

[IAutoMapperTModel, TEntity Interface](#)

[Template.Service.Core.Interface Namespace](#)

## Template Documentation



# IAutoMapperTModel, TEntityToEntityEnumerable Method

Converts an IEnumerable of Models to an IEnumerable of entities

**Namespace:** [Template.Service.Core.Interface](#)

**Assembly:** Template.Service (in Template.Service.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

```
C# Copy
    IEnumerable< TEntity > ToEntityEnumerable(
        IEnumerable< TModel > enumModel
    )
```

### Parameters

*enumModel*

Type: [System.Collections.Generic.IEnumerable< TModel >](#)

The IEnumerable of Models to be converted

### Return Value

Type: [IEnumerable< TEntity >](#)

The IEnumerable of entities

## ► See Also

### Reference

[IAutoMapperTModel, TEntity Interface](#)

[Template.Service.Core.Interface Namespace](#)



## Template Documentation



# IAutoMapper{TModel}, TEntityToEntityExpression Method

Convert an expression of a model to an expression to an Entity, be careful to make an expression of with entities properties

**Namespace:** [Template.Service.Core.Interface](#)

**Assembly:** Template.Service (in Template.Service.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

C#

Copy

```
Expression<Func< TEntity, bool>> ToEntityExpression< TEntity, TModel>(Expression<Func< TModel, bool>> predicate)
```

### Parameters

*predicate*

Type: [System.Linq.Expressions.ExpressionFunc{TModel, Boolean}](#)

**Boolean**

The expression to convert

### Return Value

Type: [ExpressionFunc{TEntity, Boolean}](#)

The TEntity expression

## ► See Also

Reference

IAutoMapperTModel, TEntity Interface  
Template.Service.Core.Interface Namespace

---

## Template Documentation



# IAutoMapper<TModel> TEntityToEntityFunction<TResult> Method

Convert a function of a model to a function of a TEntity, be careful to make an function with entities properties

**Namespace:** [Template.Service.Core.Interface](#)

**Assembly:** Template.Service (in Template.Service.dll) Version: 1.0.0.0 (1.0.0.0)

## ▲ Syntax

```
C# Copy
Func< TEntity, TResult > ToEntityFunction< TResult >(
    Func< TModel, TResult > predicate
)
◀ ▶
```

## Parameters

*predicate*

Type: [System.Func<TModel, TResult>](#)

[Missing <param name="predicate"/> documentation for

"M:Template.Service.Core.Interface.IAutoMapper`2.ToEntityFunction`1(System.Fi

## Type Parameters

*TResult*

[Missing <typeparam name="TResult"/> documentation for

"M:Template.Service.Core.Interface.IAutoMapper`2.ToEntityFunction`1(System.Fi

## Return Value

Type: `Func<TEntity, TResult>`

[Missing <returns> documentation for

"M:Template.Service.Core.Interface.IAutoMapper`2.ToEntityFunction``1(System.Func`1[[Template.Service.Core.Entity.IEntity, Template.Service.Core], [Template.Service.Core.Model.IModel, Template.Service.Core]])"

## See Also

### Reference

[IAutoMapperTModel, TEntity Interface](#)

[Template.Service.Core.Interface Namespace](#)

---

## Template Documentation



# IAutoMapperTModel, TEntityToModel Method

Converts an entity to a model

**Namespace:** [Template.Service.Core.Interface](#)

**Assembly:** Template.Service (in Template.Service.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

```
C#  
TModel ToModel(  
    TEntity entity  
)
```

[Copy](#)

### Parameters

*entity*

Type: [TEntity](#)

The entity to be converted

### Return Value

Type: [TModel](#)

The model object

## ► See Also

Reference

[IAutoMapperTModel, TEntity Interface](#)

[Template.Service.Core.Interface Namespace](#)

## Template Documentation



# IAutoMapperTModel, TEntityToModelEnumerable Method

Convert an IEnumerable of entities to an IEnumerable of models

**Namespace:** [Template.Service.Core.Interface](#)

**Assembly:** Template.Service (in Template.Service.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

```
C# Copy
IEnumerable<TModel> ToModelEnumerable(
    IEnumerable<TEntity> enumEntity
)
```

### Parameters

*enumEntity*

Type: [System.Collections.Generic.IEnumerable`1\[TEntity\]](#)

[Missing <param name="enumEntity"/> documentation for  
"M:Template.Service.Core.Interface.IAutoMapper`2.ToModelEnumerable(System.C

### Return Value

Type: [IEnumerable`1\[TModel\]](#)

The IEnumerable of models

## ► See Also

Reference

[IAutoMapperTModel, TEntity Interface](#)

## Template.Service.Core.Interface Namespace

---

## Template Documentation



# IServiceTModel Interface

Defines all the Generic Service functions

**Namespace:** [Template.Service.Core.Interface](#)

**Assembly:** Template.Service (in Template.Service.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

C#

Copy

```
public interface IService<TModel>
where TModel : BaseModel
```

### Type Parameters

*TModel*

The Model class that represents the services

The [IServiceTModel](#) type exposes the following members.

## ► Methods

	Name	Description
≡	<a href="#">Add</a>	Add an entity based on the passed model and insert it on database
≡	<a href="#">AddIEnumerableable</a>	Add an IEnumerable of entities based on the passed models IEnumerable and insert them on database

≡	<a href="#">Delete</a>	Add an entity based on the passed model and delete it on database
≡	<a href="#">DeleteBy</a>	Deletes any match with the given predicate
≡	<a href="#">Edit</a>	Add an entity based on the passed model and delete it on database
≡	<a href="#">FindBy</a>	Retuns an IEnumerable of models based on the expresion, retreives entities from the database and convert them to models
≡	<a href="#">Get</a>	Get an unique model based on the expresion, retreives the entity and convert it to model
≡	<a href="#">GetAll</a>	Get all the elements of the database, retreives entities from the database and convert them to models
≡	<a href="#">SelectWhereTResult</a>	This function allows to select the specific properties and filter by where clause on database, not on memory, in that way it improves the performace

[Top](#)

## ▲ See Also

## Reference

### [Template.Service.Core.Interface Namespace](#)

---

## Template Documentation



# IServiceTModel Methods

The [IServiceTModel](#) generic type exposes the following members.

## ▪ Methods

	Name	Description
≡	<a href="#">Add</a>	Add an entity based on the passed model and insert it on database
≡	<a href="#">AddIEnumerableable</a>	Add an IEnumerable of entities based on the passed models IEnumerable and insert them on database
≡	<a href="#">Delete</a>	Add an entity based on the passed model and delete it on database
≡	<a href="#">DeleteBy</a>	Deletes any match with the given predicate
≡	<a href="#">Edit</a>	Add an entity based on the passed model and delete it on database
≡	<a href="#">FindBy</a>	Retuns an IEnumerable of models based on the expresion, retrieves entities from the database and convert them to models

	<a href="#">Get</a>	Get an unique model based on the expresion, retrieves the entity and convert it to model
≡	<a href="#">GetAll</a>	Get all the elements of the database, retrieves entities from the database and convert them to models
≡	<a href="#">SelectWhereTResult</a>	This function allows to select the specific properties and filter by where clause on database, not on memory, in that way it improves the performace

[Top](#)

## See Also

[Reference](#)

[IServiceTModel Interface](#)

[Template.Service.Core.Interface Namespace](#)

## Template Documentation



# IServiceTModelAdd Method

Add an entity based on the passed model and insert it on database

**Namespace:** [Template.Service.Core.Interface](#)

**Assembly:** Template.Service (in Template.Service.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

C#

Copy

```
void Add(  
    TModel model  
)
```

### Parameters

*model*

Type: [TModel](#)

The model to convert in a entity

## ► See Also

[Reference](#)

[IServiceTModel Interface](#)

[Template.Service.Core.Interface Namespace](#)

## Template Documentation



# IServiceTModelAddIEnumetable Method

Add an IEnumerable of entities based on the passed models  
IEnumerable amd insert them on database

**Namespace:** [Template.Service.Core.Interface](#)

**Assembly:** Template.Service (in Template.Service.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

C#

[Copy](#)

```
void AddIEnumetable(  
    IEnumerable<TModel> models  
)
```

### Parameters

*models*

Type: [System.Collections.Generic.IEnumerable<TModel>](#)

The model IEnumerable to convert an entity IEnumerable

## ► See Also

[Reference](#)

[IServiceTModel Interface](#)

[Template.Service.Core.Interface Namespace](#)

## Template Documentation



# IServiceTModel/Delete Method

Add an entity based on the passed model and delete it on database

**Namespace:** [Template.Service.Core.Interface](#)

**Assembly:** Template.Service (in Template.Service.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

C#

Copy

```
void Delete(  
    TModel model  
)
```

### Parameters

*model*

Type: [TModel](#)

The model to convert in a entity

## ► See Also

Reference

[IServiceTModel Interface](#)

[Template.Service.Core.Interface Namespace](#)

## Template Documentation



# IServiceTModel/DeleteBy Method

Deletes any match with the given predicate

**Namespace:** [Template.Service.Core.Interface](#)

**Assembly:** Template.Service (in Template.Service.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

C#

Copy

```
int DeleteBy(  
    Expression<Func<TModel, bool>> predicate  
)
```

### Parameters

*predicate*

Type: [System.Linq.Expressions.Expression<Func<TModel, Boolean>>](#)

The predicate to filter the result

### Return Value

Type: [Int32](#)

The number of affected rows entity

## ► See Also

Reference

[IServiceTModel Interface](#)

[Template.Service.Core.Interface Namespace](#)

## Template Documentation



# IServiceTModel/Edit Method

Add an entity based on the passed model and delete it on database

**Namespace:** [Template.Service.Core.Interface](#)

**Assembly:** Template.Service (in Template.Service.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

C#

Copy

```
void Edit(  
    TModel model  
)
```

### Parameters

*model*

Type: [TModel](#)

[Missing <param name="model"/> documentation for  
"M:Template.Service.Core.Interface(IService`1.Edit(`0)"]

## ► See Also

Reference

[IServiceTModel Interface](#)

[Template.Service.Core.Interface Namespace](#)

## Template Documentation



# IServiceTModel/FindBy Method

Returns an IEnumerable of models based on the expression, retrieves entities from the database and convert them to models

**Namespace:** [Template.Service.Core.Interface](#)

**Assembly:** Template.Service (in Template.Service.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

C#

[Copy](#)

```
IEnumerable<TModel> FindBy(  
    Expression<Func<TModel, bool>> predicate  
)
```

### Parameters

*predicate*

Type: [System.Linq.Expressions.ExpressionFuncTModel](#),

[Boolean](#)

The expression to filter the selection

### Return Value

Type: [IEnumerableTModel](#)

The IEnumerable of models based on the predicate

## ► See Also

[Reference](#)

[IServiceTModel Interface](#)

[Template.Service.Core.Interface Namespace](#)

## Template Documentation



# IServiceTModel/Get Method

Get an unique model based on the expresion, retreives the entity and convert it to model

**Namespace:** [Template.Service.Core.Interface](#)

**Assembly:** Template.Service (in Template.Service.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

C#

[Copy](#)

```
TModel Get(  
    Expression<Func<TModel, bool>> predicate  
)
```

### Parameters

*predicate*

Type: [System.Linq.Expressions.ExpressionFunc<TModel, Boolean>](#)

**Boolean**

The expression to filter the selection

### Return Value

Type: [TModel](#)

The model based on the predicate

## ► See Also

[Reference](#)

[IServiceTModel Interface](#)

[Template.Service.Core.Interface Namespace](#)

## Template Documentation



# IServiceTModel GetAll Method

Get all the elements of the database, retrieves entities from the database and convert them to models

**Namespace:** [Template.Service.Core.Interface](#)

**Assembly:** Template.Service (in Template.Service.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

C#

[Copy](#)

```
IEnumerable<TModel> GetAll()
```

Return Value

Type: [IEnumerableTModel](#)

The IEnumerable of models

## ► See Also

Reference

[IServiceTModel Interface](#)

[Template.Service.Core.Interface Namespace](#)

## Template Documentation



# IService{TModel}SelectWhere{TResult} Method

This function allows to select the specific properties and filter by where clause on database, not on memory, in that way it improves the performance

**Namespace:** [Template.Service.Core.Interface](#)

**Assembly:** Template.Service (in Template.Service.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

C#

[Copy](#)

```
IEnumerable<TResult> SelectWhere<TResult>(
    Func<TModel, TResult> selector,
    Expression<Func<TModel, bool>> predicate
)
```

## Parameters

*selector*

Type: [System.Func{TModel, TResult}](#)

The properties to select

*predicate*

Type: [System.Linq.Expressions.ExpressionFunc{TModel, Boolean}](#)

[Boolean](#)

The where clause

## Type Parameters

*TResult*

The select result type

**Return Value**

Type: [IEnumerable<TResult>](#)

The IEnumerable of the result

## See Also

Reference

[IServiceTModel Interface](#)

[Template.Service.Core.Interface Namespace](#)

---

## Template Documentation



# Template.Service.Interface Namespace

## ▪ Interfaces

Interface	Description
<a href="#">IMoviesCategoryService</a>	
<a href="#">IMoviesService</a>	Interface to declare all the UserProfile logic functions

## Template Documentation



# IMoviesCategoryService Interface

[Missing <summary> documentation for "T:Template.Service.Interface.IMoviesCategoryService"]

**Namespace:** [Template.Service.Interface](#)

**Assembly:** Template.Service (in Template.Service.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

```
C#  
public interface IMoviesCategoryService : IService  
{  
    // Methods  
}
```

The [IMoviesCategoryService](#) type exposes the following members.

## ► Methods

	Name	Description
≡	<a href="#">Add</a>	Add an entity based on the passed model and insert it on database (Inherited from <a href="#">IServiceTModel</a> .)
≡	<a href="#">AddIEnumerableable</a>	Add an IEnumerable of entities based on the passed models IEnumerable and insert them on database

		(Inherited from <a href="#">IServiceTModel</a> .)
	<a href="#">Delete</a>	Add an entity based on the passed model and delete it on database (Inherited from <a href="#">IServiceTModel</a> .)
	<a href="#">DeleteBy</a>	Deletes any match with the given predicate (Inherited from <a href="#">IServiceTModel</a> .)
	<a href="#">Edit</a>	Add an entity based on the passed model and delete it on database (Inherited from <a href="#">IServiceTModel</a> .)
	<a href="#">FindBy</a>	Retuns an IEnumerable of models based on the expresion, retreives entities from the database and convert them to models (Inherited from <a href="#">IServiceTModel</a> .)
	<a href="#">Get</a>	Get an unique model based on the expresion, retreives the entity and convert it to model (Inherited from <a href="#">IServiceTModel</a> .)
	<a href="#">GetAll</a>	Get all the elements of the database, retreives entities from the database and

convert them to models  
(Inherited from  
[IServiceTModel](#).)

---



### [SelectWhereTResult](#)

This function allows to select the specific properties and filter by where clause on database, not on memory, in that way it improves the performance  
(Inherited from  
[IServiceTModel](#).)

---

[Top](#)

## ► See Also

Reference

[Template.Service.Interface Namespace](#)

---

## Template Documentation



# IMoviesCategoryService

## Methods

The [IMoviesCategoryService](#) type exposes the following members.

### ▪ Methods

Name	Description
<a href="#">Add</a>	Add an entity based on the passed model and insert it on database (Inherited from <a href="#">IServiceTModel</a> .)
<a href="#">AddIEnumerable</a>	Add an IEnumerable of entities based on the passed models IEnumerable and insert them on database (Inherited from <a href="#">IServiceTModel</a> .)
<a href="#">Delete</a>	Add an entity based on the passed model and delete it on database (Inherited from <a href="#">IServiceTModel</a> .)
<a href="#">DeleteBy</a>	Deletes any match with the given predicate (Inherited from <a href="#">IServiceTModel</a> .)

≡	Edit	Add an entity based on the passed model and delete it on database (Inherited from <a href="#">IServiceTModel</a> .)
≡	FindBy	Retuns an IEnumerable of models based on the expresion, retreives entities from the database and convert them to models (Inherited from <a href="#">IServiceTModel</a> .)
≡	Get	Get an unique model based on the expresion, retreives the entity and convert it to model (Inherited from <a href="#">IServiceTModel</a> .)
≡	GetAll	Get all the elements of the database, retreives entities from the database and convert them to models (Inherited from <a href="#">IServiceTModel</a> .)
≡	SelectWhereTResult	This function allows to select the specific properties and filter by where clause on database, not on memory, in that way it improves the performace (Inherited from <a href="#">IServiceTModel</a> .)

[Top](#)

## ◀ See Also

Reference

[IMoviesCategoryService Interface](#)  
[Template.Service.Interface Namespace](#)

---

## Template Documentation



# IMoviesService Interface

Interface to declare all the UserProfile logic functions

**Namespace:** [Template.Service.Interface](#)

**Assembly:** Template.Service (in Template.Service.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

```
C# Copy
public interface IMoviesService : IService<Movies>
```

The [IMoviesService](#) type exposes the following members.

## ► Methods

	Name	Description
≡	<a href="#">Add</a>	Add an entity based on the passed model and insert it on database (Inherited from <a href="#">IServiceTModel</a> .)
≡	<a href="#">AddIEnumerable</a>	Add an IEnumerable of entities based on the passed models IEnumerable and insert them on database (Inherited from <a href="#">IServiceTModel</a> .)
≡	<a href="#">Delete</a>	Add an entity based on the

		passed model and delete it on database (Inherited from <a href="#">IServiceTModel</a> .)
	<a href="#">DeleteBy</a>	Deletes any match with the given predicate (Inherited from <a href="#">IServiceTModel</a> .)
	<a href="#">Edit</a>	Add an entity based on the passed model and delete it on database (Inherited from <a href="#">IServiceTModel</a> .)
	<a href="#">FindBy</a>	Retuns an IEnumerable of models based on the expresion, retreives entities from the database and convert them to models (Inherited from <a href="#">IServiceTModel</a> .)
	<a href="#">Get</a>	Get an unique model based on the expresion, retreives the entity and convert it to model (Inherited from <a href="#">IServiceTModel</a> .)
	<a href="#">GetAll</a>	Get all the elements of the database, retreives entities from the database and convert them to models (Inherited from <a href="#">IServiceTModel</a> .)



**SelectWhereTResult** This function allows to select the specific properties and filter by where clause on database, not on memory, in that way it improves the performace (Inherited from [IServiceTModel](#).)

---

[Top](#)

## ▲ See Also

Reference

[Template.Service.Interface Namespace](#)

---

## Template Documentation



# IMoviesService Methods

The [IMoviesService](#) type exposes the following members.

## ▪ Methods

Name	Description
<a href="#">Add</a>	Add an entity based on the passed model and insert it on database (Inherited from <a href="#">IServiceTModel</a> .)
<a href="#">AddIEnumerable</a>	Add an IEnumerable of entities based on the passed models IEnumerable and insert them on database (Inherited from <a href="#">IServiceTModel</a> .)
<a href="#">Delete</a>	Add an entity based on the passed model and delete it on database (Inherited from <a href="#">IServiceTModel</a> .)
<a href="#">DeleteBy</a>	Deletes any match with the given predicate (Inherited from <a href="#">IServiceTModel</a> .)
<a href="#">Edit</a>	Add an entity based on the passed model and delete it

on database  
(Inherited from  
[IServiceTModel](#).)

≡	<a href="#">FindBy</a>	Retuns an IEnumerable of models based on the expresion, retreives entities from the database and convert them to models (Inherited from <a href="#">IServiceTModel</a> .)
≡	<a href="#">Get</a>	Get an unique model based on the expresion, retreives the entity and convert it to model (Inherited from <a href="#">IServiceTModel</a> .)
≡	<a href="#">GetAll</a>	Get all the elements of the database, retreives entities from the database and convert them to models (Inherited from <a href="#">IServiceTModel</a> .)
≡	<a href="#">SelectWhereTResult</a>	This function allows to select the specific properties and filter by where clause on database, not on memory, in that way it improves the performace (Inherited from <a href="#">IServiceTModel</a> .)

[Top](#)

## ▲ See Also

Reference

[IMoviesService Interface](#)

[Template.Service.Interface Namespace](#)

---

## Template Documentation



# Template.Web Namespace

## ► Classes

	Class	Description
	<a href="#">FilterConfig</a>	Used to configure all the custom filters
	<a href="#">MvcApplication</a>	Main class of the application
	<a href="#">RouteConfig</a>	Class to configure all the routes of the app

## Template Documentation



# FilterConfig Class

Used to configure all the custom filters

## ► Inheritance Hierarchy

[SystemObject](#) [Template.WebFilterConfig](#)

**Namespace:** [Template.Web](#)

**Assembly:** Template.Web (in Template.Web.dll) Version: 1.0.0.0  
(1.0.0.0)

## ► Syntax

C#

[Copy](#)

```
public class FilterConfig
```

The [FilterConfig](#) type exposes the following members.

## ► Constructors

	Name	Description
≡	<a href="#">FilterConfig</a>	Initializes a new instance of the <a href="#">FilterConfig</a> class

[Top](#)

## ► Methods

	Name	Description
≡ S	<a href="#">RegisterGlobalFilters</a>	Add all the custom filters to the GlobalFilters

[Top](#)

## ◀ See Also

Reference

[Template.Web Namespace](#)

---

## Template Documentation



# FilterConfig Constructor

Initializes a new instance of the [FilterConfig](#) class

**Namespace:** [Template.Web](#)

**Assembly:** Template.Web (in Template.Web.dll) Version: 1.0.0.0  
(1.0.0.0)

## ► Syntax

C#

Copy

```
public FilterConfig()
```

## ► See Also

[Reference](#)

[FilterConfig Class](#)

[Template.Web Namespace](#)

## Template Documentation



# FilterConfig Methods

The [FilterConfig](#) type exposes the following members.

## ▪ Methods

Name	Description
 	<a href="#">RegisterGlobalFilters</a> Add all the custom filters to the GlobalFilters

[Top](#)

## ▪ See Also

[Reference](#)

[FilterConfig Class](#)

[Template.Web Namespace](#)

## Template Documentation



# FilterConfig.RegisterGlobalFilters Method

Add all the custom filters to the GlobalFilters

**Namespace:** [Template.Web](#)

**Assembly:** Template.Web (in Template.Web.dll) Version: 1.0.0.0  
(1.0.0.0)

## ► Syntax

```
C# Copy
public static void RegisterGlobalFilters(
    GlobalFilterCollection filters
)
```

## Parameters

*filters*

Type: [GlobalFilterCollection](#)

The GlobalFilter collection of the MVC app

## ► See Also

Reference

[FilterConfig Class](#)

[Template.Web Namespace](#)

## Template Documentation



# MvcApplication Class

Main class of the application

## ► Inheritance Hierarchy

```
SystemObject System.Web.HttpApplication  
Template.Web.MvcApplication
```

**Namespace:** [Template.Web](#)

**Assembly:** Template.Web (in Template.Web.dll) Version: 1.0.0.0  
(1.0.0.0)

## ► Syntax

C#

[Copy](#)

```
public class MvcApplication : HttpApplication
```

The [MvcApplication](#) type exposes the following members.

## ► Constructors

	Name	Description
≡	<a href="#">MvcApplication</a>	Initializes a new instance of the <a href="#">MvcApplication</a> class

[Top](#)

## ► Methods

	Name	Description
💡	<a href="#">Application_Error</a>	Execeutes on every exception



[Application\\_Start](#) The main function of the application

---

[Top](#)

## ► See Also

[Reference](#)

[Template.Web Namespace](#)

---

## Template Documentation



# MvcApplication Constructor

Initializes a new instance of the [MvcApplication](#) class

**Namespace:** [Template.Web](#)

**Assembly:** Template.Web (in Template.Web.dll) Version: 1.0.0.0  
(1.0.0.0)

## ► Syntax

C#

Copy

```
public MvcApplication()
```

## ► See Also

[Reference](#)

[MvcApplication Class](#)

[Template.Web Namespace](#)

## Template Documentation



# MvcApplication Methods

The [MvcApplication](#) type exposes the following members.

## ▪ Methods

	Name	Description
♪	<a href="#">Application_Error</a>	Excecutes on every exception
♪	<a href="#">Application_Start</a>	The main function of the application

[Top](#)

## ▪ See Also

Reference

[MvcApplication Class](#)

[Template.Web Namespace](#)

## Template Documentation



# MvcApplicationApplication\_Error Method

Execeutes on every exception

**Namespace:** [Template.Web](#)

**Assembly:** Template.Web (in Template.Web.dll) Version: 1.0.0.0  
(1.0.0.0)

## ► Syntax

```
C# Copy
protected void Application_Error(
    Object sender,
    EventArgs e
)
```

### Parameters

*sender*

Type: [SystemObject](#)

The object that triggered the event

*e*

Type: [SystemEventArgs](#)

The event properties

## ► See Also

[Reference](#)

[MvcApplication Class](#)

[Template.Web Namespace](#)

## Template Documentation



# MvcApplicationApplication\_Start Method

The main function of the application

**Namespace:** [Template.Web](#)

**Assembly:** Template.Web (in Template.Web.dll) Version: 1.0.0.0  
(1.0.0.0)

## ► Syntax

```
C# _____ Copy
protected void Application_Start()
```

## ► See Also

[Reference](#)

[MvcApplication Class](#)

[Template.Web Namespace](#)

## Template Documentation



# RouteConfig Class

Class to configure all the routes of the app

## ► Inheritance Hierarchy

[SystemObject](#) [Template.WebRouteConfig](#)

**Namespace:** [Template.Web](#)

**Assembly:** Template.Web (in Template.Web.dll) Version: 1.0.0.0  
(1.0.0.0)

## ► Syntax

C#

[Copy](#)

```
public class RouteConfig
```

The `RouteConfig` type exposes the following members.

## ► Constructors

	Name	Description
≡	<a href="#">RouteConfig</a>	Initializes a new instance of the <code>RouteConfig</code> class

[Top](#)

## ► Methods

	Name	Description
≡ S	<a href="#">RegisterRoutes</a>	Register all the routes for the app

[Top](#)

## ◀ See Also

Reference

[Template.Web Namespace](#)

---

## Template Documentation



# RouteConfig Constructor

Initializes a new instance of the [RouteConfig](#) class

**Namespace:** [Template.Web](#)

**Assembly:** Template.Web (in Template.Web.dll) Version: 1.0.0.0  
(1.0.0.0)

## ► Syntax

C#

Copy

```
public RouteConfig()
```

## ► See Also

[Reference](#)

[RouteConfig Class](#)

[Template.Web Namespace](#)

## Template Documentation



# RouteConfig Methods

The [RouteConfig](#) type exposes the following members.

## ▪ Methods

	Name	Description
 	<a href="#">RegisterRoutes</a>	Register all the routes for the app

[Top](#)

## ▪ See Also

[Reference](#)

[RouteConfig Class](#)

[Template.Web Namespace](#)

## Template Documentation



# RouteConfig.RegisterRoutes Method

Register all the routes for the app

**Namespace:** [Template.Web](#)

**Assembly:** Template.Web (in Template.Web.dll) Version: 1.0.0.0  
(1.0.0.0)

## ► Syntax

C#

[Copy](#)

```
public static void RegisterRoutes(  
    RouteCollection routes  
)
```

## Parameters

*routes*

Type: [System.Web.Routing.RouteCollection](#)

The main routes collection object to map the routes

## ► See Also

Reference

[RouteConfig Class](#)

[Template.Web Namespace](#)

## Template Documentation



# Template.Web.App\_Start Namespace

## ► Classes

	Class	Description
	<a href="#">BundleConfig</a>	Class to configure all the bundles of the MVC app

## Template Documentation



# BundleConfig Class

Class to configure all the bundles of the MVC app

## ► Inheritance Hierarchy

[SystemObject](#) [Template.Web.App\\_StartBundleConfig](#)

**Namespace:** [Template.Web.App\\_Start](#)

**Assembly:** Template.Web (in Template.Web.dll) Version: 1.0.0.0  
(1.0.0.0)

## ► Syntax

C#

[Copy](#)

```
public class BundleConfig
```

The `BundleConfig` type exposes the following members.

## ► Constructors

	Name	Description
≡	<a href="#">BundleConfig</a>	Initializes a new instance of the <code>BundleConfig</code> class

[Top](#)

## ► Methods

	Name	Description
≡ S	<a href="#">RegisterBundles</a>	Register all the bundles for styles and scripts

[Top](#)

## ◀ See Also

Reference

[Template.Web.App\\_Start Namespace](#)

---

## Template Documentation



# BundleConfig Constructor

Initializes a new instance of the [BundleConfig](#) class

**Namespace:** [Template.Web.App\\_Start](#)

**Assembly:** Template.Web (in Template.Web.dll) Version: 1.0.0.0  
(1.0.0.0)

## ► Syntax

C#

Copy

```
public BundleConfig()
```

## ► See Also

Reference

[BundleConfig Class](#)

[Template.Web.App\\_Start Namespace](#)

## Template Documentation



# BundleConfig Methods

The [BundleConfig](#) type exposes the following members.

## ▪ Methods

	Name	Description
 	<a href="#">RegisterBundles</a>	Register all the bundles for styles and scripts

[Top](#)

## ▪ See Also

Reference

[BundleConfig Class](#)

[Template.Web.App\\_Start Namespace](#)

## Template Documentation



# BundleConfigRegisterBundles Method

Register all the bundles for styles and scripts

**Namespace:** [Template.Web.App\\_Start](#)

**Assembly:** Template.Web (in Template.Web.dll) Version: 1.0.0.0  
(1.0.0.0)

## ► Syntax

C#

[Copy](#)

```
public static void RegisterBundles(  
    BundleCollection bundles  
)
```

### Parameters

*bundles*

Type: [BundleCollection](#)

Used to register the bundles

## ► See Also

Reference

[BundleConfig Class](#)

[Template.Web.App\\_Start Namespace](#)

## Template Documentation



# Template.Web.Commons.Security Namespace

## ► Classes

	Class	Description
	<a href="#">MembershipConfig</a>	Manage all the authentication configuration for the MVC app
	<a href="#">URLModifyPrevent</a>	Used to avoid the SQL injection by the URL, when the URL is modify manually, redirect to a Malicious request page

## Template Documentation



# MembershipConfig Class

Manage all the authentication configuration for the MVC app

## ► Inheritance Hierarchy

[SystemObject](#) [Template.Web.Commons.SecurityMembershipConfig](#)

**Namespace:** [Template.Web.Commons.Security](#)

**Assembly:** Template.Web (in Template.Web.dll) Version: 1.0.0.0  
(1.0.0.0)

## ► Syntax

C#

[Copy](#)

```
public class MembershipConfig
```

The [MembershipConfig](#) type exposes the following members.

## ► Constructors

	Name	Description
≡	<a href="#">MembershipConfig</a>	Initializes a new instance of the <a href="#">MembershipConfig</a> class

[Top](#)

## ► Methods

	Name	Description
≡ S	<a href="#">Initialize</a>	Initialized the membership configuration with the database name, the userprofile

table name, the user profile primary key name and the username description column name

---

[Top](#)

## ▲ See Also

Reference

[Template.Web.Commons.Security Namespace](#)

---

## Template Documentation



# MembershipConfig Constructor

Initializes a new instance of the [MembershipConfig](#) class

**Namespace:** [Template.Web.Commons.Security](#)

**Assembly:** Template.Web (in Template.Web.dll) Version: 1.0.0.0  
(1.0.0.0)

## ► Syntax

C#

Copy

```
public MembershipConfig()
```

## ► See Also

[Reference](#)

[MembershipConfig Class](#)

[Template.Web.Commons.Security Namespace](#)

## Template Documentation



# MembershipConfig Methods

The [MembershipConfig](#) type exposes the following members.

## ▪ Methods

	Name	Description
 	<a href="#">Initialize</a>	Initialized the membership configuration with the database name, the userprofile table name, the user profile primary key name and the username description column name

[Top](#)

## ▪ See Also

[Reference](#)

[MembershipConfig Class](#)

[Template.Web.Commons.Security Namespace](#)

## Template Documentation



# MembershipConfigInitialize Method

Initialized the membership configuration with the database name, the userprofile table name, the user profile primery key name and the username description column name

**Namespace:** [Template.Web.Commons.Security](#)

**Assembly:** Template.Web (in Template.Web.dll) Version: 1.0.0.0  
(1.0.0.0)

## ► Syntax

C#

[Copy](#)

```
public static void Initialize()
```

## ► See Also

Reference

[MembershipConfig Class](#)

[Template.Web.Commons.Security Namespace](#)

## Template Documentation



# URLModifyPrevent Class

Used to avoid the SQL injection by the URL, when the URL is modify manually, redirect to a Malicious request page

## ► Inheritance Hierarchy

```
SystemObject SystemAttribute  
FilterAttribute  
ActionFilterAttribute  
Template.Web.Commons.SecurityURLModifyPrevent
```

**Namespace:** [Template.Web.Commons.Security](#)

**Assembly:** Template.Web (in Template.Web.dll) Version: 1.0.0.0  
(1.0.0.0)

## ► Syntax

C#

[Copy](#)

```
[AttributeUsageAttribute(AttributeTargets.Class)]  
public class URLModifyPrevent : ActionFilterAttri
```

The [URLModifyPrevent](#) type exposes the following members.

## ► Constructors

	Name	Description
≡	<a href="#">URLModifyPrevent</a>	Initializes a new instance of the URLModifyPrevent class

[Top](#)

## ◀ Methods

Name	Description
 <a href="#">OnActionExecuting</a>	Verify the manipulation of the URL on execution. (Overrides <a href="#">ActionFilterAttribute.OnActionExecuting</a> )

[Top](#)

## ◀ See Also

Reference

[Template.Web.Commons.Security Namespace](#)

## Template Documentation



# URLModifyPrevent Constructor

Initializes a new instance of the [URLModifyPrevent](#) class

**Namespace:** [Template.Web.Commons.Security](#)

**Assembly:** Template.Web (in Template.Web.dll) Version: 1.0.0.0  
(1.0.0.0)

## ► Syntax

C#

Copy

```
public URLModifyPrevent()
```

## ► See Also

[Reference](#)

[URLModifyPrevent Class](#)

[Template.Web.Commons.Security Namespace](#)

## Template Documentation



# URLModifyPrevent Methods

The [URLModifyPrevent](#) type exposes the following members.

## ▪ Methods

Name	Description
 <a href="#">OnActionExecuting</a>	Verify the manipulation of the URL on execution. (Overrides <a href="#">ActionFilterAttribute.OnActionExecuting</a> )

[Top](#)

## ▪ See Also

Reference

[URLModifyPrevent Class](#)

[Template.Web.Commons.Security Namespace](#)

## Template Documentation



# URLModifyPreventOnActionExecuting Method

Verify the manipulation of the URL on every action request

**Namespace:** [Template.Web.Commons.Security](#)

**Assembly:** Template.Web (in Template.Web.dll) Version: 1.0.0.0  
(1.0.0.0)

## ► Syntax

```
C# Copy
public override void OnActionExecuting(
    ActionExecutingContext filterContext
)
```

### Parameters

*filterContext*

Type: [ActionExecutingContext](#)

[Missing <param name="filterContext"/> documentation for  
"M:Template.Web.Commons.Security.URLModifyPrevent.OnActionExecuting(Sys...")]

### Implements

[IActionFilter.OnActionExecuting\(ActionExecutingContext\)](#)

## ► See Also

[Reference](#)

[URLModifyPrevent Class](#)

[Template.Web.Commons.Security Namespace](#)

## Template Documentation



# Template.Web.Commons.Utils Namespace

## ► Classes

	Class	Description
	<a href="#">DropDownHelper</a>	Generic class to generate all the DropDownList
	<a href="#">JsonHelper</a>	Class to help with the JSON responses
	<a href="#">SessionManager</a>	class to manage the sessions
	<a href="#">ViewExtensions</a>	Class to declare all the extension functions for the MVC app

## Template Documentation



# DropDownHelper Class

Generic class to generate all the DropDownList

## ► Inheritance Hierarchy

[SystemObject](#) [Template.Web.Commons.Utils](#)[DropDownHelper](#)

**Namespace:** [Template.Web.Commons.Utils](#)

**Assembly:** Template.Web (in Template.Web.dll) Version: 1.0.0.0  
(1.0.0.0)

## ► Syntax

C#

Copy

```
public static class DropDownHelper
```

## ► Methods

	Name	Description
≡ S	<a href="#">GetDropdown(IEnumerableString, Int32)</a>	Generate a list of SelectListItem using a normal list of String as key and value
≡ S	<a href="#">GetDropdownT(IEnumerableT, FuncT, String, FuncT, String, Int32)</a>	Generate a list of SelectListItem
≡ S	<a href="#">GetDropdownT(IEnumerableT,</a>	Generate a

`FuncT, String, FuncT, String,  
FuncT, String, Char, Int32)`

list of  
SelectListItem  
using two  
attributes of  
the model as  
text to display  
it on the  
dropdown

---

[Top](#)

## ► See Also

Reference

[Template.Web.Commons.Utils Namespace](#)

---

## Template Documentation



# DropDownHelper Methods

## ► Methods

	Name	Description
≡  	<a href="#">GetDropdown(IEnumerableString, Int32)</a>	Generate a list of SelectListItem using a normal list of String as key and value
≡  	<a href="#">GetDropdownT(IEnumerableT, FuncT, String, FuncT, String, Int32)</a>	Generate a list of SelectListItem
≡  	<a href="#">GetDropdownT(IEnumerableT, FuncT, String, FuncT, String, FuncT, String, Char, Int32)</a>	Generate a list of SelectListItem using two attributes of the model as text to display it on the dropdown

[Top](#)

## ► See Also

[Reference](#)

[DropDownHelper Class](#)

## Template.Web.Commons.Utils Namespace

---

## Template Documentation



# DropDownHelper.GetDropdown Method

## ▪ Overload List

	Name	Description
≡  	<code>GetDropdown(IEnumerableString, Int32)</code>	Generate a list of SelectListItem using a normal list of String as key and value
≡  	<code>GetDropdownT(IEnumerableT, FuncT, String, FuncT, String, Int32)</code>	Generate a list of SelectListItem
≡  	<code>GetDropdownT(IEnumerableT, FuncT, String, FuncT, String, FuncT, String, Char, Int32)</code>	Generate a list of SelectListItem using two attributes of the model as text to display it on the dropdown

[Top](#)

## ▪ See Also

## Reference

[DropDownHelper Class](#)

[Template.Web.Commons.Utils Namespace](#)

---

## Template Documentation



# DropDownHelper.GetDropdown Method (IEnumerableString, Int32)

Generate a list of SelectListItem using a normal list of String as key and value

**Namespace:** [Template.Web.Commons.Utils](#)

**Assembly:** Template.Web (in Template.Web.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

```
C# Copy
public static IEnumerable<SelectListItem> GetDrop
    IEnumerable<string> listdata,
    int selected = 0
)
```

### Parameters

*listdata*

Type: [System.Collections.Generic.IEnumerableString](#)

The source String list

*selected (Optional)*

Type: [System.Int32](#)

Index of which element whatns to be selected of the begining, default 0

### Return Value

Type: [IEnumerableSelectListItem](#)

A list of SelectedListIem ready to use as a source of a dropdown

with Razor

## See Also

Reference

[DropDownHelper Class](#)

[GetDropdown Overload](#)

[Template.Web.Commons.Utils Namespace](#)

---

## Template Documentation



# DropDownHelper.GetDropdown $T$

## Method (IEnumerable $T$ , Func $T$ , String, Func $T$ , String, Int32)

Generate a list of SelectListItem

**Namespace:** [Template.Web.Commons.Utils](#)

**Assembly:** Template.Web (in Template.Web.dll) Version: 1.0.0.0 (1.0.0.0)

### ► Syntax

```
C# Copy
public static IEnumerable<SelectListItem> GetDrop
    IEnumerable<T> listModel,
    Func<T, string> key,
    Func<T, string> text,
    int selected = 0
)
```

### Parameters

*listModel*

Type: [System.Collections.Generic.IEnumerable \$T\$](#)

The list source of the model

*key*

Type: [SystemFunc \$T\$ , String](#)

A function to get the property of the model to use as a key of the dropdown

*text*

Type: [SystemFunc \$T\$ , String](#)

A function to get the property of the model to use as a value text

of the dropdown  
**selected (Optional)**

Type: [SystemInt32](#)

Index of which element whatns to be selected of the begining,  
default 0

## Type Parameters

*T*

The Model type for generate the list

## Return Value

Type: [IEnumerableSelectListItem](#)

A list of SelectedListItem ready to use as a source of a dropdown  
with Razor

## See Also

### Reference

[DropDownHelper Class](#)

[GetDropdown Overload](#)

[Template.Web.Commons.Utils Namespace](#)

## Template Documentation



# DropDownHelper.GetDropdownT Method (IEnumerableT, FuncT, String, FuncT, String, FuncT, String, Char, Int32)

Generate a list of SelectListItem using two attributes of the model as text to display it on the dropdown

**Namespace:** [Template.Web.Commons.Utils](#)

**Assembly:** Template.Web (in Template.Web.dll) Version: 1.0.0.0  
(1.0.0.0)

## ► Syntax

```
C# Copy
public static IEnumerable<SelectListItem> GetDropdownT(
    IEnumerable<T> listModel,
    Func<T, string> key,
    Func<T, string> firstText,
    Func<T, string> secondText,
    char concatCharacter,
    int selected = 0
)
```

## Parameters

*listModel*

Type: [System.Collections.Generic.IEnumerableT](#)

The list source of the model

*key*

Type: [SystemFuncT, String](#)

A function to get the property of the model to use as a key of the dropdown

*firstText*

Type: [SystemFunc<T, String>](#)

A function to get the property of the model to use as the first part of the text of the dropdown

*secondText*

Type: [SystemFunc<T, String>](#)

A function to get the property of the model to use as the second part of the text of the dropdown

*concatCharacter*

Type: [SystemChar](#)

A character to concat both text as one to display it on the dropdown

*selected (Optional)*

Type: [SystemInt32](#)

Index of which element whatns to be selected of the begining, default 0

## Type Parameters

*T*

The Model type for generate the list

## Return Value

Type: [IEnumerable<SelectListItem>](#)

A list of SelectListItem ready to use as a source of a dropdown with Razor

## See Also

### Reference

[DropDownHelper Class](#)

[GetDropdown Overload](#)

[Template.Web.Commons.Utils Namespace](#)

## Template Documentation



# JsonHelper Class

Class to help with the JSON responses

## ► Inheritance Hierarchy

[SystemObject](#) [Template.Web.Commons.Utils.JsonHelper](#)

**Namespace:** [Template.Web.Commons.Utils](#)

**Assembly:** Template.Web (in Template.Web.dll) Version: 1.0.0.0  
(1.0.0.0)

## ► Syntax

C#

[Copy](#)

```
public static class JsonHelper
```

## ► Methods

	Name	Description
≡  S	<a href="#">Fail</a>	Returns a Fail response (isSuccess = false)
≡  S	<a href="#">Fail(String)</a>	Returns a fail response with the given text (isSuccess = false, responseText = responseText)
≡  S	<a href="#">Fail(String, Object)</a>	Returns a fail Response with the given text and serialize object (isSuccess = false, responseText = responseText, date = data)
≡  S		

	<b>Success</b>	Returns a Success response ( isSuccess = true)
 	<b>Success(String)</b>	Returns a Success response with the given text ( isSuccess = true, responseText = responseText)
 	<b>Success(String, Object)</b>	Returns a Success Response with the given text and serialize object

---

[Top](#)

## ► See Also

Reference

[Template.Web.Commons.Utils Namespace](#)

---

## Template Documentation



# JsonHelper Methods

## ↳ Methods

	Name	Description
≡ 	<a href="#">Fail</a>	Returns a Fail response ( isSuccess = false )
≡ 	<a href="#">Fail(String)</a>	Returns a fail response with the given text ( isSuccess = false, responseText = responseText )
≡ 	<a href="#">Fail(String, Object)</a>	Returns a fail Response with the given text and serialize object ( isSuccess = false, responseText = responseText, date = data )
≡ 	<a href="#">Success</a>	Returns a Success response ( isSuccess = true )
≡ 	<a href="#">Success(String)</a>	Returns a Success response with the given text ( isSuccess = true, responseText = responseText )
≡ 	<a href="#">Success(String, Object)</a>	Returns a Success Response with the given text and serialize object

[Top](#)

## ↳ See Also

## Reference

[JsonHelper Class](#)

[Template.Web.Commons.Utils Namespace](#)

---

## Template Documentation



# JsonHelperFail Method

## ▪ Overload List

	Name	Description
≡ 	<a href="#">Fail</a>	Returns a Fail response (isSuccess = false)
≡ 	<a href="#">Fail(String)</a>	Returns a fail response with the given text (isSuccess = false, responseText = responseText)
≡ 	<a href="#">Fail(String, Object)</a>	Returns a fail Response with the given text and serialize object (isSuccess = false, responseText = responseText, date = data)

[Top](#)

## ▪ See Also

[Reference](#)

[JsonHelper Class](#)

[Template.Web.Commons.Utils Namespace](#)

## Template Documentation



# JsonHelperFail Method

Returns a Fail response (isSuccess = false)

**Namespace:** [Template.Web.Commons.Utils](#)

**Assembly:** Template.Web (in Template.Web.dll) Version: 1.0.0.0  
(1.0.0.0)

## ► Syntax

C#

Copy

```
public static JsonResult Fail()
```

Return Value

Type: [JsonResult](#)

The JsonResult object

## ► See Also

[Reference](#)

[JsonHelper Class](#)

[Fail Overload](#)

[Template.Web.Commons.Utils Namespace](#)

## Template Documentation



# JsonHelperFail Method (String)

Returns a fail response with the given text (isSuccess = false, responseText = responseText)

**Namespace:** [Template.Web.Commons.Utils](#)

**Assembly:** Template.Web (in Template.Web.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

C#

[Copy](#)

```
public static JsonResult Fail(  
    string responseText  
)
```

### Parameters

*responseText*

Type: [System.String](#)

The text to returns

### Return Value

Type: [JsonResult](#)

The JsonResult object

## ► See Also

[Reference](#)

[JsonHelper Class](#)

[Fail Overload](#)

[Template.Web.Commons.Utils Namespace](#)

## Template Documentation



# JsonHelperFail Method (String, Object)

Returns a fail Response with the given text and serialize object  
(isSuccess = false, responseText = responseText, date = data)

**Namespace:** [Template.Web.Commons.Utils](#)

**Assembly:** Template.Web (in Template.Web.dll) Version: 1.0.0.0  
(1.0.0.0)

## ► Syntax

C#

[Copy](#)

```
public static JsonResult Fail(  
    string responseText,  
    Object data  
)
```

## Parameters

*responseText*

Type: [SystemString](#)

The text to return

*data*

Type: [SystemObject](#)

The object to serialize

## Return Value

Type: [JsonResult](#)

The JsonResult object

## ► See Also

## Reference

[JsonHelper Class](#)

[Fail Overload](#)

[Template.Web.Commons.Utils Namespace](#)

---

## Template Documentation



# JsonHelperSuccess Method

## ▪ Overload List

	Name	Description
≡  S	<a href="#">Success</a>	Returns a Success response (isSuccess = true)
≡  S	<a href="#">Success(String)</a>	Returns a Success response with the given text (isSuccess = true, responseText = responseText)
≡  S	<a href="#">Success(String, Object)</a>	Returns a Success Response with the given text and serialize object

[Top](#)

## ▪ See Also

[Reference](#)

[JsonHelper Class](#)

[Template.Web.Commons.Utils Namespace](#)

## Template Documentation



# JsonHelperSuccess Method

Returns a Success response (isSuccess = true)

**Namespace:** [Template.Web.Commons.Utils](#)

**Assembly:** Template.Web (in Template.Web.dll) Version: 1.0.0.0  
(1.0.0.0)

## ► Syntax

C#

Copy

```
public static JsonResult Success()
```

**Return Value**

Type: [JsonResult](#)

The JsonResult object

## ► See Also

[Reference](#)

[JsonHelper Class](#)

[Success Overload](#)

[Template.Web.Commons.Utils Namespace](#)

## Template Documentation



# JsonHelperSuccess Method (String)

Returns a Success response with the given text (isSuccess = true, responseText = responseText)

**Namespace:** [Template.Web.Commons.Utils](#)

**Assembly:** Template.Web (in Template.Web.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

C#

[Copy](#)

```
public static JsonResult Success(  
    string responseText  
)
```

### Parameters

*responseText*

Type: [System.String](#)

The text to returns

### Return Value

Type: [JsonResult](#)

The JsonResult object

## ► See Also

### Reference

[JsonHelper Class](#)

[Success Overload](#)

[Template.Web.Commons.Utils Namespace](#)



## Template Documentation



# JsonHelperSuccess Method (String, Object)

Returns a Success Response with the given text and serialize object

**Namespace:** [Template.Web.Commons.Utils](#)

**Assembly:** Template.Web (in Template.Web.dll) Version: 1.0.0.0  
(1.0.0.0)

## ► Syntax

C#

[Copy](#)

```
public static JsonResult Success(  
    string responseText,  
    Object data  
)
```

### Parameters

*responseText*

Type: [SystemString](#)

[Missing <param name="responseText"/> documentation for  
"M:Template.Web.Commons.Utils.JsonHelper.Success(System.String,System.Obj

*data*

Type: [SystemObject](#)

[Missing <param name="data"/> documentation for  
"M:Template.Web.Commons.Utils.JsonHelper.Success(System.String,System.Obj

### Return Value

Type: [JsonResult](#)

The JsonResult object

## ► See Also

## Reference

[JsonHelper Class](#)

[Success Overload](#)

[Template.Web.Commons.Utils Namespace](#)

---

## Template Documentation



# SessionManager Class

class to manage the sessions

## ► Inheritance Hierarchy

[SystemObject](#) [Template.Web.Commons.UtilsSessionManager](#)

**Namespace:** [Template.Web.Commons.Utils](#)

**Assembly:** Template.Web (in Template.Web.dll) Version: 1.0.0.0  
(1.0.0.0)

## ► Syntax

C#

[Copy](#)

```
public class SessionManager
```

The [SessionManager](#) type exposes the following members.

## ► Constructors

	Name	Description
≡	<a href="#">SessionManager</a>	Initializes a new instance of the <a href="#">SessionManager</a> class

[Top](#)

## ► Methods

	Name	Description
≡ S	<a href="#">ClearSession</a>	method that clear all the keys in the session



[GetListSessionT](#) method that return the key in a list , from the current session or create one if not exist



[RemoveSession](#) method to remove a key from the current session

---

[Top](#)

## See Also

Reference

[Template.Web.Commons.Utils Namespace](#)

---

## Template Documentation



# SessionManager Constructor

Initializes a new instance of the [SessionManager](#) class

**Namespace:** [Template.Web.Commons.Utils](#)

**Assembly:** Template.Web (in Template.Web.dll) Version: 1.0.0.0  
(1.0.0.0)

## ► Syntax

C#

Copy

```
public SessionManager()
```

## ► See Also

Reference

[SessionManager Class](#)

[Template.Web.Commons.Utils Namespace](#)

## Template Documentation



# SessionManager Methods

The [SessionManager](#) type exposes the following members.

## ▪ Methods

	Name	Description
 	<a href="#">ClearSession</a>	method that clear all the keys in the session
 	<a href="#">GetListSessionT</a>	method that return the key in a list , from the current session or create one if not exist
 	<a href="#">RemoveSession</a>	method to remove a key from the current session

[Top](#)

## ▪ See Also

[Reference](#)

[SessionManager Class](#)

[Template.Web.Commons.Utils Namespace](#)

## Template Documentation



# SessionManagerClearSession Method

method that clear all the keys in the session

**Namespace:** [Template.Web.Commons.Utils](#)

**Assembly:** Template.Web (in Template.Web.dll) Version: 1.0.0.0  
(1.0.0.0)

## ► Syntax

C#

[Copy](#)

```
public static void ClearSession()
```

## ► See Also

[Reference](#)

[SessionManager Class](#)

[Template.Web.Commons.Utils Namespace](#)

## Template Documentation



# SessionManagerGetListSession<T> Method

method that return the key in a list , from the current session or create one if not exist

**Namespace:** [Template.Web.Commons.Utils](#)

**Assembly:** Template.Web (in Template.Web.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

C#

Copy

```
public static List<T> GetListSession<T>(  
    string key  
)
```

### Parameters

*key*

Type: [System.String](#)  
key name

### Type Parameters

*T*

type of the session key

### Return Value

Type: [List<T>](#)

[Missing <returns> documentation for  
"M:Template.Web.Commons.Utils.SessionManager.GetListSession`1(System.String)"]

## ▲ See Also

Reference

[SessionManager Class](#)

[Template.Web.Commons.Utils Namespace](#)

---

## Template Documentation



# SessionManagerRemoveSession Method

method to remove a key from the current session

**Namespace:** [Template.Web.Commons.Utils](#)

**Assembly:** Template.Web (in Template.Web.dll) Version: 1.0.0.0  
(1.0.0.0)

## ► Syntax

```
C#  
public static void RemoveSession(  
    string key  
)
```

[Copy](#)

## Parameters

*key*

Type: [System.String](#)

session key

## ► See Also

Reference

[SessionManager Class](#)

[Template.Web.Commons.Utils Namespace](#)

## Template Documentation



# ViewExtensions Class

Class to declare all the extension functions for the MVC app

## ► Inheritance Hierarchy

[SystemObject](#) [Template.Web.Commons.Utils](#) **ViewExtensions**

**Namespace:** [Template.Web.Commons.Utils](#)

**Assembly:** Template.Web (in Template.Web.dll) Version: 1.0.0.0  
(1.0.0.0)

## ► Syntax

C#

[Copy](#)

```
public static class ViewExtensions
```

The **ViewExtensions** type exposes the following members.

## ► Methods

	Name	Description
 	<a href="#">RenderToString</a>	Returns the PartialView render as String

[Top](#)

## ► See Also

[Reference](#)

[Template.Web.Commons.Utils Namespace](#)

## Template Documentation



# ViewExtensions Methods

The [ViewExtensions](#) type exposes the following members.

## ▪ Methods

	Name	Description
 	<a href="#">RenderToString</a>	Returns the PartialView render as String

[Top](#)

## ▪ See Also

Reference

[ViewExtensions Class](#)

[Template.Web.Commons.Utils Namespace](#)

## Template Documentation



# ViewExtensions.ToString Method

Returns the PartialView render as String

**Namespace:** [Template.Web.Commons.Utils](#)

**Assembly:** Template.Web (in Template.Web.dll) Version: 1.0.0.0  
(1.0.0.0)

## ► Syntax

```
C# Copy
public static string RenderToString(
    this PartialViewResult partialView
)
```

### Parameters

*partialView*

Type: [PartialViewResult](#)

[Missing <param name="partialView"/> documentation for  
"M:Template.Web.Commons.Utils.ViewExtensions.RenderToString(System.Web.M

### Return Value

Type: [String](#)

The String result

### Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [PartialViewResult](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

## ▲ See Also

Reference

[ViewExtensions Class](#)

[Template.Web.Commons.Utils Namespace](#)

---

## Template Documentation



# Template.Web.CompositionRoot Namespace

## ► Classes

	Class	Description
	<a href="#">CompositionRootConfig</a>	A Composition Root is unique location in an application where modules are composed together. Is used to configure all the modules of the app
	<a href="#">UnityConfig</a>	Configures the Unity dependency injection container

## Template Documentation



# CompositionRootConfig Class

A Composition Root is unique location in an application where modules are composed together. Is used to configure all the modules of the app

## ► Inheritance Hierarchy

[SystemObject](#) [Template.Web.CompositionRootCompositionRootConfig](#)

**Namespace:** [Template.Web.CompositionRoot](#)

**Assembly:** Template.Web (in [Template.Web.dll](#)) Version: 1.0.0.0  
(1.0.0.0)

## ► Syntax

C#

Copy

```
public class CompositionRootConfig
```

The [CompositionRootConfig](#) type exposes the following members.

## ► Constructors

Name	Description
 <a href="#">CompositionRootConfig</a>	Initializes a new instance of the <a href="#">CompositionRootConfig</a> class

[Top](#)

## ► Methods

Name	Description
------	-------------



[Init](#)

Initializes the Composition Root

---

[Top](#)

## ▲ See Also

[Reference](#)

[Template.Web.CompositionRoot Namespace](#)

---

## Template Documentation



# CompositionRootConfig Constructor

Initializes a new instance of the [CompositionRootConfig](#) class

**Namespace:** [Template.Web.CompositionRoot](#)

**Assembly:** Template.Web (in Template.Web.dll) Version: 1.0.0.0  
(1.0.0.0)

## ► Syntax

C#

[Copy](#)

```
public CompositionRootConfig()
```

## ► See Also

[Reference](#)

[CompositionRootConfig Class](#)

[Template.Web.CompositionRoot Namespace](#)

## Template Documentation



# CompositionRootConfig Methods

The [CompositionRootConfig](#) type exposes the following members.

## ▪ Methods

	Name	Description
 	<a href="#">Init</a>	Initializes the Composition Root

[Top](#)

## ▪ See Also

[Reference](#)

[CompositionRootConfig Class](#)

[Template.Web.CompositionRoot Namespace](#)

## Template Documentation



# CompositionRootConfigInit Method

Initializes the Composition Root

**Namespace:** [Template.Web.CompositionRoot](#)

**Assembly:** Template.Web (in Template.Web.dll) Version: 1.0.0.0  
(1.0.0.0)

## ► Syntax

```
C#  
public static void Init()
```

[Copy](#)

## ► See Also

[Reference](#)

[CompositionRootConfig Class](#)

[Template.Web.CompositionRoot Namespace](#)

## Template Documentation



# UnityConfig Class

Configures the Unity dependency injection container

## ► Inheritance Hierarchy

[SystemObject](#) [Template.Web.CompositionRootUnityConfig](#)

**Namespace:** [Template.Web.CompositionRoot](#)

**Assembly:** Template.Web (in Template.Web.dll) Version: 1.0.0.0  
(1.0.0.0)

## ► Syntax

C#

[Copy](#)

```
internal static class UnityConfig
```

The [UnityConfig](#) type exposes the following members.

## ► Methods

	Name	Description
 	<a href="#">Init</a>	Initializes Unity
 	<a href="#">RegisterTypes</a>	Automatically registers the type mappings with the Unity container by conventions.

[Top](#)

## ► See Also

[Reference](#)

## Template.Web.CompositionRoot Namespace

---

## Template Documentation



# UnityConfig Methods

The [UnityConfig](#) type exposes the following members.

## ▪ Methods

	Name	Description
≡ S	<a href="#">Init</a>	Initializes Unity
≡ S	<a href="#">RegisterTypes</a>	Automatically registers the type mappings with the Unity container by conventions.

[Top](#)

## ▪ See Also

[Reference](#)

[UnityConfig Class](#)

[Template.Web.CompositionRoot Namespace](#)

## Template Documentation



# UnityConfigInit Method

Initializes Unity

**Namespace:** [Template.Web.CompositionRoot](#)

**Assembly:** Template.Web (in Template.Web.dll) Version: 1.0.0.0  
(1.0.0.0)

## ► Syntax

C#

Copy

```
public static void Init()
```

## ► See Also

Reference

[UnityConfig Class](#)

[Template.Web.CompositionRoot Namespace](#)

## Template Documentation



# UnityConfig.RegisterType Method

Automatically registers the type mappings with the Unity container by conventions.

**Namespace:** [Template.Web.CompositionRoot](#)

**Assembly:** Template.Web (in Template.Web.dll) Version: 1.0.0.0 (1.0.0.0)

## ► Syntax

C#

[Copy](#)

```
public static void RegisterTypes(  
    IUnityContainer container  
)
```

### Parameters

*container*

Type: **IUnityContainer**

The unity container to configure.

## ► See Also

[Reference](#)

[UnityConfig Class](#)

[Template.Web.CompositionRoot Namespace](#)

## Template Documentation



# Template.Web.Controllers Namespace

## ► Classes

	Class	Description
	<a href="#">HomeController</a>	Class to manage all the request of 'Home'

## Template Documentation



# HomeController Class

Class to manage all the request of 'Home'

## ► Inheritance Hierarchy

```
SystemObject ControllerBase  
Controller  
Template.Web.ControllersHomeController
```

**Namespace:** [Template.Web.Controllers](#)

**Assembly:** Template.Web (in Template.Web.dll) Version: 1.0.0.0  
(1.0.0.0)

## ► Syntax

C#

[Copy](#)

```
public class HomeController : Controller
```

The [HomeController](#) type exposes the following members.

## ► Constructors

	Name	Description
≡	<a href="#">HomeController</a>	Initializes a new instance of the <a href="#">HomeController</a> class

[Top](#)

## ► Methods

	Name	Description
≡		

## Create

≡	<a href="#">Create(MoviesModel)</a>
≡	<a href="#">Delete</a>
≡	<a href="#">Details</a>
≡	<a href="#">Edit(Int32)</a>
≡	<a href="#">Edit(MoviesModel)</a>
≡	<a href="#">Index</a> Render the Index view on GET request

[Top](#)

## Fields

Name	Description
≡	<a href="#">_moviesCategoryService</a>
≡	<a href="#">_moviesService</a>

[Top](#)

## See Also

Reference

[Template.Web.Controllers Namespace](#)

## Template Documentation



# HomeController Constructor

Initializes a new instance of the [HomeController](#) class

**Namespace:** [Template.Web.Controllers](#)

**Assembly:** Template.Web (in Template.Web.dll) Version: 1.0.0.0  
(1.0.0.0)

## ► Syntax

C#

Copy

```
public HomeController(  
    IMoviesService moviesService,  
    IMoviesCategoryService moviesCategoryService  
)
```

### Parameters

*moviesService*

Type: [Template.Service.InterfaceIMoviesService](#)  
[Missing <param name="moviesService"/> documentation for  
"M:Template.Web.Controllers.HomeController.#ctor(Template.Service.Interface.IM

*moviesCategoryService*

Type: [Template.Service.InterfaceIMoviesCategoryService](#)  
[Missing <param name="moviesCategoryService"/> documentation for  
"M:Template.Web.Controllers.HomeController.#ctor(Template.Service.Interface.IM

## ► See Also

Reference

[HomeController Class](#)

[Template.Web.Controllers Namespace](#)

## Template Documentation



# HomeController Methods

The [HomeController](#) type exposes the following members.

## ▪ Methods

Name	Description
<a href="#">Create</a>	
<a href="#">Create(MoviesModel)</a>	
<a href="#">Delete</a>	
<a href="#">Details</a>	
<a href="#">Edit(Int32)</a>	
<a href="#">Edit(MoviesModel)</a>	
<a href="#">Index</a>	Render the Index view on GET request

[Top](#)

## ▪ See Also

[Reference](#)

[HomeController Class](#)

[Template.Web.Controllers Namespace](#)

## Template Documentation



# HomeControllerCreate Method

## ► Overload List

Name	Description
 <a href="#">Create</a>	
 <a href="#">Create(MoviesModel)</a>	

[Top](#)

## ► See Also

[Reference](#)

[HomeController Class](#)

[Template.Web.Controllers Namespace](#)

## Template Documentation



# HomeControllerCreate Method

[Missing <summary> documentation for  
"M:Template.Web.Controllers.HomeController.Create"]

**Namespace:** [Template.Web.Controllers](#)

**Assembly:** Template.Web (in Template.Web.dll) Version: 1.0.0.0  
(1.0.0.0)

## ► Syntax

C#

[Copy](#)

```
[HttpGetAttribute]  
public ViewResult Create()
```

**Return Value**

Type: [ViewResult](#)

[Missing <returns> documentation for  
"M:Template.Web.Controllers.HomeController.Create"]

## ► See Also

**Reference**

[HomeController Class](#)

[Create Overload](#)

[Template.Web.Controllers Namespace](#)

## Template Documentation



# HomeControllerCreate Method (MoviesModel)

[Missing <summary> documentation for  
"M:Template.Web.Controllers.HomeController.Create(Template.Core.Model.MoviesModel)"]

**Namespace:** [Template.Web.Controllers](#)

**Assembly:** Template.Web (in Template.Web.dll) Version: 1.0.0.0  
(1.0.0.0)

## ► Syntax

C#

[Copy](#)

```
[HttpPostAttribute]
public RedirectToRouteResult Create(
    MoviesModel model
)
```

## Parameters

*model*

Type: **MoviesModel**

[Missing <param name="model"/> documentation for  
"M:Template.Web.Controllers.HomeController.Create(Template.Core.Model.Movie

## Return Value

Type: [RedirectToRouteResult](#)

[Missing <returns> documentation for  
"M:Template.Web.Controllers.HomeController.Create(Template.Core.Model.MoviesMod

## ► See Also

### Reference

HomeController Class

Create Overload

Template.Web.Controllers Namespace

---

## Template Documentation



# HomeControllerDelete Method

[Missing <summary> documentation for  
"M:Template.Web.Controllers.HomeController.Delete(System.Int32)"]

**Namespace:** [Template.Web.Controllers](#)

**Assembly:** Template.Web (in Template.Web.dll) Version: 1.0.0.0  
(1.0.0.0)

## ► Syntax

C#

[Copy](#)

```
[HttpGetAttribute]
public RedirectToRouteResult Delete(
    int id
)
```

## Parameters

*id*

Type: [SystemInt32](#)

[Missing <param name="id"/> documentation for  
"M:Template.Web.Controllers.HomeController.Delete(System.Int32)"]

## Return Value

Type: [RedirectToRouteResult](#)

[Missing <returns> documentation for  
"M:Template.Web.Controllers.HomeController.Delete(System.Int32)"]

## ► See Also

[Reference](#)

[HomeController Class](#)

[Template.Web.Controllers Namespace](#)



## Template Documentation



# HomeControllerDetails Method

[Missing <summary> documentation for  
"M:Template.Web.Controllers.HomeController.Details(System.Int32)"]

**Namespace:** [Template.Web.Controllers](#)

**Assembly:** Template.Web (in Template.Web.dll) Version: 1.0.0.0  
(1.0.0.0)

## ► Syntax

```
C#  
[HttpGetAttribute]  
public ViewResult Details(  
    int id  
)
```

[Copy](#)

## Parameters

*id*

Type: [SystemInt32](#)

[Missing <param name="id"/> documentation for  
"M:Template.Web.Controllers.HomeController.Details(System.Int32)"]

## Return Value

Type: [ViewResult](#)

[Missing <returns> documentation for  
"M:Template.Web.Controllers.HomeController.Details(System.Int32)"]

## ► See Also

[Reference](#)

[HomeController Class](#)

[Template.Web.Controllers Namespace](#)



## Template Documentation



# HomeControllerEdit Method

## ► Overload List

Name	Description
 <a href="#">Edit(Int32)</a>	
 <a href="#">Edit(MoviesModel)</a>	

[Top](#)

## ► See Also

[Reference](#)

[HomeController Class](#)

[Template.Web.Controllers Namespace](#)

## Template Documentation



# HomeControllerEdit Method (Int32)

[Missing <summary> documentation for  
"M:Template.Web.Controllers.HomeController.Edit(System.Int32)"]

**Namespace:** [Template.Web.Controllers](#)

**Assembly:** Template.Web (in Template.Web.dll) Version: 1.0.0.0  
(1.0.0.0)

## ► Syntax

C#

[Copy](#)

```
[HttpGetAttribute]
public ViewResult Edit(
    int id
)
```

## Parameters

*id*

Type: [SystemInt32](#)

[Missing <param name="id"/> documentation for  
"M:Template.Web.Controllers.HomeController.Edit(System.Int32)"]

## Return Value

Type: [ViewResult](#)

[Missing <returns> documentation for  
"M:Template.Web.Controllers.HomeController.Edit(System.Int32)"]

## ► See Also

### Reference

[HomeController Class](#)

[Edit Overload](#)

[Template.Web.Controllers Namespace](#)

---

## Template Documentation



# HomeControllerEdit Method (MoviesModel)

[Missing <summary> documentation for  
"M:Template.Web.Controllers.HomeController.Edit(Template.Core.Model.MoviesModel)"]

**Namespace:** [Template.Web.Controllers](#)

**Assembly:** Template.Web (in Template.Web.dll) Version: 1.0.0.0  
(1.0.0.0)

## ► Syntax

C#

[Copy](#)

```
[HttpPostAttribute]
public RedirectToRouteResult Edit(
    MoviesModel model
)
```

## Parameters

*model*

Type: **MoviesModel**

[Missing <param name="model"/> documentation for

"M:Template.Web.Controllers.HomeController.Edit(Template.Core.Model.MoviesM

## Return Value

Type: [RedirectToRouteResult](#)

[Missing <returns> documentation for

"M:Template.Web.Controllers.HomeController.Edit(Template.Core.Model.MoviesModel)"

## ► See Also

### Reference

[HomeController Class](#)

[Edit Overload](#)

[Template.Web.Controllers Namespace](#)

---

## Template Documentation



# HomeControllerIndex Method

Render the Index view on GET request

**Namespace:** [Template.Web.Controllers](#)

**Assembly:** Template.Web (in Template.Web.dll) Version: 1.0.0.0  
(1.0.0.0)

## ► Syntax

```
C#  
[HttpGetAttribute]  
public ViewResult Index()
```

[Copy](#)

Return Value

Type: [ViewResult](#)

The ViewResult object response

## ► See Also

Reference

[HomeController Class](#)

[Template.Web.Controllers Namespace](#)

## Template Documentation



# HomeController Fields

## ↳ Fields

	Name	Description
	<a href="#">_moviesCategoryService</a>	
	<a href="#">_moviesService</a>	

[Top](#)

## ↳ See Also

[Reference](#)

[HomeController Class](#)

[Template.Web.Controllers Namespace](#)

## Template Documentation



# HomeController.\_moviesCategoryService

## Field

[Missing <summary> documentation for  
"F:Template.Web.Controllers.HomeController.\_moviesCategoryService"]

**Namespace:** [Template.Web.Controllers](#)

**Assembly:** Template.Web (in Template.Web.dll) Version: 1.0.0.0  
(1.0.0.0)

### ► Syntax

C#

[Copy](#)

```
private readonly IMoviesCategoryService _moviesCategoryService;
```

Field Value

Type: [IMoviesCategoryService](#)

### ► See Also

Reference

[HomeController Class](#)

[Template.Web.Controllers Namespace](#)

## Template Documentation



# HomeController.\_moviesService Field

[Missing <summary> documentation for  
"F:Template.Web.Controllers.HomeController.\_moviesService"]

**Namespace:** [Template.Web.Controllers](#)

**Assembly:** Template.Web (in Template.Web.dll) Version: 1.0.0.0  
(1.0.0.0)

## ► Syntax

C#

[Copy](#)

```
private readonly IMoviesService _moviesService
```

Field Value

Type: [IMoviesService](#)

## ► See Also

[Reference](#)

[HomeController Class](#)

[Template.Web.Controllers Namespace](#)