## MongoRepository Namespace

Provides classes to use MongoDB with a repository pattern.

### Classes

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong><img src="image1.png" alt="Image" /></strong></td>
<td><strong>CollectionName</strong></td>
</tr>
<tr>
<td></td>
<td>Attribute used to annotate Entities with to override mongo collection name. By default, when this attribute is not specified, the classname will be used.</td>
</tr>
<tr>
<td><strong><img src="image2.png" alt="Image" /></strong></td>
<td><strong>Entity</strong></td>
</tr>
<tr>
<td></td>
<td>Abstract Entity for all the BusinessEntities.</td>
</tr>
<tr>
<td><strong><img src="image3.png" alt="Image" /></strong></td>
<td><strong>MongoRepositoryT</strong></td>
</tr>
<tr>
<td></td>
<td>Deals with entities in MongoDB.</td>
</tr>
<tr>
<td><strong><img src="image4.png" alt="Image" /></strong></td>
<td><strong>MongoRepositoryT, TKey</strong></td>
</tr>
<tr>
<td></td>
<td>Deals with entities in MongoDB.</td>
</tr>
<tr>
<td><strong><img src="image5.png" alt="Image" /></strong></td>
<td><strong>MongoRepositoryManagerT</strong></td>
</tr>
<tr>
<td></td>
<td>Deals with the collections of entities in MongoDB. This class tries to hide as much MongoDB-</td>
</tr>
</tbody>
</table>
specific details as possible but it's not 100% yet. It is a very thin wrapper around most methods on MongoDB's MongoCollection objects.

**MongoRepositoryManager**

*MongoRepositoryManagerT, TKey*

Deals with the collections of entities in MongoDB. This class tries to hide as much MongoDB-specific details as possible but it's not 100% yet. It is a very thin wrapper around most methods on MongoDB's MongoCollection objects.

## Interfaces

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEntity</td>
<td>&quot;Default&quot; Entity interface.</td>
</tr>
<tr>
<td>IEntity&lt;TKey&gt;</td>
<td>Generic Entity interface.</td>
</tr>
<tr>
<td>IRepositoryT</td>
<td>IRepository definition.</td>
</tr>
<tr>
<td>IRepositoryT, TKey</td>
<td>IRepository definition.</td>
</tr>
</tbody>
</table>
| IRepositotyManagerT, TKey | IRepositotyManagerT-definition.
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>IRepositotyManagerT</td>
<td>IRepositotyManagerT-definition.</td>
</tr>
</tbody>
</table>

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
CollectionName Class

Attribute used to annotate Entities with to override mongo collection name. By default, when this attribute is not specified, the classname will be used.

Inheritance Hierarchy

- System
  - Object
  - SystemAttribute
    - MongoRepository.CollectionName

Namespace: MongoRepository
Version: 1.6.11.0 (1.6.11.0)

Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
<th>C++</th>
<th>F#</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```csharp
public class CollectionName : Attribute
```

The `CollectionName` type exposes the following members.

Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>*</code> CollectionName</td>
<td>Initializes a new instance of the CollectionName class attribute with the desired name.</td>
</tr>
</tbody>
</table>

Properties
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Gets the name of the collection.</td>
</tr>
<tr>
<td>Typeld</td>
<td>When implemented in a derived class, gets a unique identifier for this attribute. (Inherited from Attribute.)</td>
</tr>
</tbody>
</table>

**Top**

## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Returns a value that indicates whether this instance is equal to a specified object. (Inherited from Attribute.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Returns the hash code for this instance. (Inherited from Attribute.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>IsDefaultAttribute</td>
<td>When overridden in a derived class, indicates whether the value of this instance is the default value for the derived class.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Match</td>
<td>When overridden in a derived class, returns a value that indicates whether this instance equals a specified object. (Inherited from Attribute.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

**See Also**

- Reference
- MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
CollectionName Constructor

Initializes a new instance of the CollectionName class attribute with the desired name.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

**C#**

```csharp
public CollectionName(
    string value
)
```

### Parameters

**value**

Type: `System.String`  
Name of the collection.

### See Also

Reference

- [CollectionName Class](#)
- [MongoRepository Namespace](#)

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
CollectionName Properties

The *CollectionName* type exposes the following members.

## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Gets the name of the collection.</td>
</tr>
<tr>
<td>Typeld</td>
<td>When implemented in a derived class, gets a unique identifier for this Attribute. (Inherited from Attribute.)</td>
</tr>
</tbody>
</table>

See Also

Reference

- CollectionName Class
- MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
**CollectionName Name Property**

Gets the name of the collection.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
<th>C++</th>
<th>F#</th>
</tr>
</thead>
</table>
| ```csharp
public virtual string Name { get; }
``` |

### Property Value

Type: **String**  
The name of the collection.

### See Also

Reference  
**CollectionName Class**  
**MongoRepository Namespace**

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
The `CollectionName` type exposes the following members.

## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Equals</code></td>
<td>Returns a value that indicates whether this instance is equal to a specified object. (Inherited from <code>Attribute</code>.)</td>
</tr>
<tr>
<td><code>Finalize</code></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>GetHashCode</code></td>
<td>Returns the hash code for this instance. (Inherited from <code>Attribute</code>.)</td>
</tr>
<tr>
<td><code>GetType</code></td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>IsDefaultValue</code></td>
<td>When overridden in a derived class, indicates whether the value of this instance is the default value for the derived class. (Inherited from <code>Attribute</code>.)</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Match</td>
<td>When overridden in a derived class, returns a value that indicates whether this instance equals a specified object.</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object.</td>
</tr>
</tbody>
</table>

**See Also**

**Reference**

CollectionName Class
MongoRepository Namespace
Entity Class

Abstract Entity for all the BusinessEntities.

Inheritance Hierarchy

System Object  MongoRepositoryEntity

Namespace: MongoRepository
Version: 1.6.11.0 (1.6.11.0)

Syntax

C#  VB  C++  F#

```csharp
[SerializableAttribute]
public abstract class Entity : IEntity<string>
```

The Entity type exposes the following members.

Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>🌐 Entity</td>
<td>Initializes a new instance of the Entity class</td>
</tr>
</tbody>
</table>

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>📖 Id</td>
<td>Gets or sets the id for this object (the</td>
</tr>
</tbody>
</table>
primary record for an entity).

## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Allows an object to try to free resources and perform other cleanup</td>
</tr>
<tr>
<td></td>
<td>operations before it is reclaimed by garbage collection.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as the default hash function.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <strong>Type</strong> of the current instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Creates a shallow copy of the current <strong>Object</strong>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a string that represents the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <strong>Object</strong>.)</td>
</tr>
</tbody>
</table>
See Also

Reference
MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
Entity Constructor

Initializes a new instance of the Entity class

Namespace: MongoRepository
Version: 1.6.11.0 (1.6.11.0)

Syntax

C#  VB  C++  F#

```csharp
protected Entity()
```

See Also

Reference
Entity Class
MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
Entity Properties

The `Entity` type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>Gets or sets the id for this object (the primary record for an entity).</td>
</tr>
</tbody>
</table>

See Also

Reference
- Entity Class
- MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl
Send comments on this topic to Rob Janssen
EntityId Property

Gets or sets the id for this object (the primary record for an entity).

Namespace: MongoRepository
Version: 1.6.11.0 (1.6.11.0)

▲ Syntax

```csharp
public virtual string Id { get; set; }
```

Property Value
Type: String
The id for this object (the primary record for an entity).

Implements
IEntityTKeyId

▲ See Also

Reference
Entity Class
MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
Entity Methods

The **Entity** type exposes the following members.

## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <em>Object.</em>)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <em>Object.</em>)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from <em>Object.</em>)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <em>Type</em> of the current instance. (Inherited from <em>Object.</em>)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current <em>Object</em>. (Inherited from <em>Object.</em>)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from <em>Object.</em>)</td>
</tr>
</tbody>
</table>
See Also

Reference
Entity Class
MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl
Send comments on this topic to Rob Janssen
IEntity Interface

"Default" Entity interface.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

```csharp
public interface IEntity : IEntity<string>
```

The `IEntity` type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| ![Image](image.png) | Id | Gets or sets the Id of the Entity.  
(Inherited from `IEntity<TKey>`) |

### Remarks

Entities are assumed to use strings for Id's.

### See Also

Reference  
MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl
Send comments on this topic to Rob Janssen
IEntity Properties

The IEntity type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>Gets or sets the Id of the Entity. (Inherited from IEntity&lt;TKey&gt;.)</td>
</tr>
</tbody>
</table>

See Also

Reference

IEntity Interface
MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
IEntity<TKey> Interface

Generic Entity interface.

Namespace: MongoRepository
Version: 1.6.11.0 (1.6.11.0)

Syntax

```
public interface IEntity<TKey>
```

Type Parameters

*TKey*

The type used for the entity's Id.

The `IEntity<TKey>` type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Image]</td>
<td>Id</td>
</tr>
</tbody>
</table>

See Also

Reference

MongoRepository Namespace
Send comments on this topic to Rob Janssen
The `IEntity<TKey>` generic type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>Gets or sets the Id of the Entity.</td>
</tr>
</tbody>
</table>

### See Also

Reference
- `IEntity<TKey> Interface`
- `MongoRepository Namespace`

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
**IEntity<TKey> Id Property**

Gets or sets the Id of the Entity.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

## Syntax

```csharp
TKey Id { get; set; }
```

**Property Value**

Type: *TKey*

Id of the Entity.

## See Also

**Reference**

- IEntity<TKey> Interface
- MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
IRepository<T> Interface

IRepository definition.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
<th>C++</th>
<th>F#</th>
</tr>
</thead>
<tbody>
<tr>
<td>public interface IRepository&lt;T&gt; : IRepository&lt;T, IQueryable&lt;T&gt;, IEnumerable&lt;T&gt;, IQueryable&lt;T&gt; where T : Object, IEntity&lt;string&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Type Parameters**

*T*  
The type contained in the repository.

The `IRepository<T>` type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![image]</td>
<td><strong>Collection</strong> Gets the Mongo collection (to perform advanced operations). (Inherited from <code>IRepository&lt;T, TKey&gt;</code>.</td>
</tr>
<tr>
<td>![image]</td>
<td><strong>ElementType</strong> Gets the type of the element(s) that are returned when the</td>
</tr>
</tbody>
</table>
expression tree associated with this instance of `IQueryable` is executed.
(Inherited from `IQueryable`.)

**Expression**
Gets the expression tree that is associated with the instance of `IQueryable`.
(Inherited from `IQueryable`.)

**Provider**
Gets the query provider that is associated with this data source.
(Inherited from `IQueryable`.)

---

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add(IEnumerable&lt;T&gt;)</td>
<td>Adds the new entities in the repository. (Inherited from <code>IRepositoryT, TKey</code>.)</td>
</tr>
<tr>
<td>Add(T)</td>
<td>Adds the new entity in the repository. (Inherited from <code>IRepositoryT, TKey</code>.)</td>
</tr>
<tr>
<td>Count</td>
<td>Counts the total entities in the repository. (Inherited from <code>IRepositoryT, TKey</code>.)</td>
</tr>
<tr>
<td>Delete(ExpressionFuncT, Boolean)</td>
<td>Deletes the entities matching the</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Delete(T)</td>
<td>Deletes the given entity.</td>
</tr>
<tr>
<td>Delete(TKey)</td>
<td>Deletes an entity from the repository by its id.</td>
</tr>
<tr>
<td>DeleteAll</td>
<td>Deletes all entities in the repository.</td>
</tr>
<tr>
<td>Exists</td>
<td>Checks if the entity exists for given predicate.</td>
</tr>
<tr>
<td>GetById</td>
<td>Returns the T by its given id.</td>
</tr>
<tr>
<td>GetEnumerater</td>
<td>Returns an enumerator that iterates through the collection.</td>
</tr>
<tr>
<td>RequestDone</td>
<td>Lets the server know that this thread is done</td>
</tr>
</tbody>
</table>

*Inherited from IRepositoryT, TKey.*
with a series of related operations.
(Inherited from IRepositoryT, TKey.)

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RequestStart</td>
<td>Lets the server know that this thread is about to begin a series of related operations that must all occur on the same connection. The return value of this method implements IDisposable and can be placed in a using statement (in which case RequestDone will be called automatically when leaving the using statement). (Inherited from IRepositoryT, TKey.)</td>
</tr>
<tr>
<td>Update(IEnumerableT)</td>
<td>Upserts the entities. (Inherited from IRepositoryT, TKey.)</td>
</tr>
<tr>
<td>Update(T)</td>
<td>Upserts an entity. (Inherited from IRepositoryT, TKey.)</td>
</tr>
</tbody>
</table>

Remarks

Entities are assumed to use strings for Id's.
See Also

Reference

MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
IRepository\textit{T} Properties

The \textit{IRepository\textit{T}} generic type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collection</td>
<td>Gets the Mongo collection (to perform advanced operations). (Inherited from \textit{IRepository\textit{T}, TKey}.)</td>
</tr>
<tr>
<td>ElementType</td>
<td>Gets the type of the element(s) that are returned when the expression tree associated with this instance of \textit{IQueryable} is executed. (Inherited from \textit{IQueryable}.)</td>
</tr>
<tr>
<td>Expression</td>
<td>Gets the expression tree that is associated with the instance of \textit{IQueryable}. (Inherited from \textit{IQueryable}.)</td>
</tr>
<tr>
<td>Provider</td>
<td>Gets the query provider that is associated with this data source. (Inherited from \textit{IQueryable}.)</td>
</tr>
</tbody>
</table>

See Also

Reference
The **IRepository<T>** generic type exposes the following members.

## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add(IEnumerable&lt;T&gt;)</td>
<td>Adds the new entities in the repository. (Inherited from IRepository&lt;T, TKey).</td>
</tr>
<tr>
<td>Add(T)</td>
<td>Adds the new entity in the repository. (Inherited from IRepository&lt;T, TKey).</td>
</tr>
<tr>
<td>Count</td>
<td>Counts the total entities in the repository. (Inherited from IRepository&lt;T, TKey).</td>
</tr>
<tr>
<td>Delete(Expression&lt;Func&lt;T, Boolean&gt;&gt;)</td>
<td>Deletes the entities matching the predicate. (Inherited from IRepository&lt;T, TKey).</td>
</tr>
<tr>
<td>Delete(T)</td>
<td>Deletes the given entity. (Inherited from IRepository&lt;T, TKey).</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Delete(TKey)</td>
<td>Deletes an entity from the repository by its id.</td>
</tr>
<tr>
<td>DeleteAll</td>
<td>Deletes all entities in the repository.</td>
</tr>
<tr>
<td>Exists</td>
<td>Checks if the entity exists for given predicate.</td>
</tr>
<tr>
<td>GetById</td>
<td>Returns the T by its given id.</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the collection.</td>
</tr>
<tr>
<td>RequestDone</td>
<td>Lets the server know that this thread is done with a series of related operations.</td>
</tr>
<tr>
<td>RequestStart</td>
<td>Lets the server know that this thread is about to begin a series of related operations</td>
</tr>
</tbody>
</table>
that must all occur on the same connection. The return value of this method implements IDisposable and can be placed in a using statement (in which case RequestDone will be called automatically when leaving the using statement). (Inherited from IRepositoryT, TKey.)

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
<th>Inherited From</th>
</tr>
</thead>
<tbody>
<tr>
<td>Update(IEnumerable T)</td>
<td>Upserts the entities.</td>
<td>IRepositoryT, TKey</td>
</tr>
<tr>
<td>Update(T)</td>
<td>Upserts an entity.</td>
<td>IRepositoryT, TKey</td>
</tr>
</tbody>
</table>

**See Also**

**Reference**
- IRepositoryT Interface
- MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
IRepository\(T, TKey\) Interface

IRepository definition.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

```
c# public interface IRepository\<T, TKey\> : IQueryable\<T\>, IEnumerable\<T\>, IQueryable, IEnumerable
where T : Object, IEntity<TKey>
```

Type Parameters

\(T\)

The type contained in the repository.

\(TKey\)

The type used for the entity's Id.

The \texttt{IRepository\(T, TKey\)} type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collection</td>
<td>Gets the Mongo collection (to perform advanced operations).</td>
</tr>
<tr>
<td>ElementType</td>
<td>Gets the type of the element(s) that are returned when the</td>
</tr>
</tbody>
</table>
expression tree associated with this instance of `IQueryable` is executed. (Inherited from `IQueryable`.)

### Expression

- Gets the expression tree that is associated with the instance of `IQueryable`. (Inherited from `IQueryable`.)

### Provider

- Gets the query provider that is associated with this data source. (Inherited from `IQueryable`.)

---

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Add(IEnumerable&lt;T&gt;)</code></td>
<td>Adds the new entities in the repository.</td>
</tr>
<tr>
<td><code>Add(T)</code></td>
<td>Adds the new entity in the repository.</td>
</tr>
<tr>
<td><code>Count</code></td>
<td>Counts the total entities in the repository.</td>
</tr>
<tr>
<td><code>Delete(ExpressionFuncT, Boolean)</code></td>
<td>Deletes the entities matching the predicate.</td>
</tr>
<tr>
<td><code>Delete(T)</code></td>
<td>Deletes the given entity.</td>
</tr>
<tr>
<td><code>Delete(TKey)</code></td>
<td>Deletes an entity from</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DeleteAll</td>
<td>Deletes all entities in the repository.</td>
</tr>
<tr>
<td>Exists</td>
<td>Checks if the entity exists for given predicate.</td>
</tr>
<tr>
<td>GetById</td>
<td>Returns the T by its given id.</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the collection. (Inherited from <code>IEnumerable&lt;T&gt;</code>)</td>
</tr>
<tr>
<td>RequestDone</td>
<td>Lets the server know that this thread is done with a series of related operations.</td>
</tr>
<tr>
<td>RequestStart</td>
<td>Lets the server know that this thread is about to begin a series of related operations that must all occur on the same connection. The return value of this method implements IDisposable and can be placed in a using statement (in which case RequestDone will be called automatically when leaving the using statement).</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td><code>Update(IEnumerable&lt;T&gt;)</code></td>
<td>Upserts the entities.</td>
</tr>
<tr>
<td><code>Update(T)</code></td>
<td>Upserts an entity.</td>
</tr>
</tbody>
</table>

See Also

Reference
MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
The `IRepository{T, TKey}` generic type exposes the following members.

## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Collection</strong></td>
<td>Gets the Mongo collection (to perform advanced operations).</td>
</tr>
<tr>
<td><strong>ElementType</strong></td>
<td>Gets the type of the element(s) that are returned when the expression tree associated with this instance of <code>IQueryable</code> is executed. (Inherited from <code>IQueryable</code>.)</td>
</tr>
<tr>
<td><strong>Expression</strong></td>
<td>Gets the expression tree that is associated with the instance of <code>IQueryable</code>. (Inherited from <code>IQueryable</code>.)</td>
</tr>
<tr>
<td><strong>Provider</strong></td>
<td>Gets the query provider that is associated with this data source. (Inherited from <code>IQueryable</code>.)</td>
</tr>
</tbody>
</table>

---

### See Also

Reference

`IRepository{T, TKey}` Interface

`MongoRepository` Namespace
**IRepository{T, TKeyCollection} Property**

Gets the Mongo collection (to perform advanced operations).

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

[C#]  
```csharp
MongoCollection{T} Collection { get; }
```

[VB]  
```vbnet
Public Property Collection As MongoCollection{T}
```

[C++]  
```cpp
MongoCollection<T>* Collection;
```

[F#]  
```fsharp
let Collection : MongoCollection<T>
```

### Property Value

Type: `MongoCollection{T}`  
The Mongo collection (to perform advanced operations).

### Remarks

One can argue that exposing this property (and with that, access to it's Database property for instance (which is a "parent")) is not the responsibility of this class. Use of this property is highly discouraged; for most purposes you can use the MongoRepositoryManager<T>

### See Also

- Reference  
  - IRepository{T, TKey Interface}  
  - MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl
Send comments on this topic to Rob Janssen
### IRepository<T, TKey> Methods

The `IRepository<T, TKey>` generic type exposes the following members.

#### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add(IEnumerable&lt;T&gt;)</td>
<td>Adds the new entities in the repository.</td>
</tr>
<tr>
<td>Add(T)</td>
<td>Adds the new entity in the repository.</td>
</tr>
<tr>
<td>Count</td>
<td>Counts the total entities in the repository.</td>
</tr>
<tr>
<td>Delete(Expression&lt;Func&lt;T, Boolean&gt;&gt;)</td>
<td>Deletes the entities matching the predicate.</td>
</tr>
<tr>
<td>Delete(T)</td>
<td>Deletes the given entity.</td>
</tr>
<tr>
<td>Delete(TKey)</td>
<td>Deletes an entity from the repository by its id.</td>
</tr>
<tr>
<td>DeleteAll</td>
<td>Deletes all entities in the repository.</td>
</tr>
<tr>
<td>Exists</td>
<td>Checks if the entity exists for given predicate.</td>
</tr>
</tbody>
</table>

---

**MongoRepository documentation**
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetById</td>
<td>Returns the T by its given id.</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the collection. (Inherited from IEnumerable.)</td>
</tr>
<tr>
<td>RequestDone</td>
<td>Lets the server know that this thread is done with a series of related operations.</td>
</tr>
<tr>
<td>RequestStart</td>
<td>Lets the server know that this thread is about to begin a series of related operations that must all occur on the same connection. The return value of this method implements IDisposable and can be placed in a using statement (in which case RequestDone will be called automatically when leaving the using statement).</td>
</tr>
<tr>
<td>Update(IEnumerable T)</td>
<td>Upserts the entities.</td>
</tr>
<tr>
<td>Update(T)</td>
<td>Upserts an entity.</td>
</tr>
</tbody>
</table>

**Top**

**See Also**
Reference

IRepositoryT, TKey Interface

MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
**IRepositoryT, TKey Add Method**

**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add(IEnumerable&lt;T&gt;)</td>
<td>Adds the new entities in the repository.</td>
</tr>
<tr>
<td>Add(T)</td>
<td>Adds the new entity in the repository.</td>
</tr>
</tbody>
</table>

**See Also**

Reference
- IRepositoryT, TKey Interface
- MongoRepository Namespace
IRepository\text{T}, TKey Add Method
(IEnumerable\text{T})

Adds the new entities in the repository.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

```csharp
void Add(
    IEnumerable\text{T} entities
)
```

### Parameters

*entities*

Type: System.Collections.Generic\text{IEnumerable}\text{T}  
The entities of type T.

### See Also

Reference  
IRepository\text{T}, TKey Interface  
Add Overload  
MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
IRepository<T, TKey> Add Method (T)

Adds the new entity in the repository.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
<th>C++</th>
<th>F#</th>
</tr>
</thead>
<tbody>
<tr>
<td>T Add(T entity)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Parameters

**entity**

Type: **T**  
The entity to add.

### Return Value

Type: **T**  
The added entity including its new ObjectId.

### See Also

**Reference**  
IRepository<T, TKey> Interface  
Add Overload  
MongoRepository Namespace
IRepository<T, TKey> Count Method

Counts the total entities in the repository.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

```csharp
long Count()
```

### Return Value

Type: Int64  
Count of entities in the repository.

### See Also

Reference  
IRepository<T, TKey> Interface  
MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
## IRepository<T, TKey> Delete Method

### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delete(Expression&lt;Func&lt;T, Boolean&gt;)</td>
<td>Deletes the entities matching the predicate.</td>
</tr>
<tr>
<td>Delete(T)</td>
<td>Deletes the given entity.</td>
</tr>
<tr>
<td>Delete(TKey)</td>
<td>Deletes an entity from the repository by its id.</td>
</tr>
</tbody>
</table>

### See Also

**Reference**
- IRepository<T, TKey> Interface
- MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
IRepository<T, TKey> Delete Method (Expression<Func<T, Boolean>>)

Deletes the entities matching the predicate.

Namespace: MongoRepository
Version: 1.6.11.0 (1.6.11.0)

**Syntax**

```csharp
void Delete(
    Expression<Func<T, bool>> predicate
)
```

**Parameters**

**predicate**
Type: System.Linq.Expressions.Expression<Func<T, Boolean>>
The expression.

**See Also**

Reference
IRepository<T, TKey> Interface
Delete Overload
MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl
Send comments on this topic to Rob Janssen
IRepository\text{T}, TKey\text{Delete Method}\ (\text{T})

Deletes the given entity.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

```c#
void Delete(
    T entity
)
```

### Parameters

- **entity**  
  - **Type:** \text{T}  
  - The entity to delete.

### See Also

- **Reference**  
  - IRepository\text{T}, TKey Interface  
  - Delete Overload  
  - MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
**IRepository{T, TKey}Delete Method (TKey)**

Deletes an entity from the repository by its id.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

```csharp
void Delete(
    TKey id
)
```

**Parameters**

**id**

Type: *TKey*  
The entity’s id.

### See Also

**Reference**

- IRepository{T, TKey} Interface  
- Delete Overload  
- MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
IRepository<T, TKey> DeleteAll Method

Deletes all entities in the repository.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

## Syntax

```csharp
void DeleteAll()
```

## See Also

**Reference**
- IRepository<T, TKey> Interface  
- MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
IRepository\text{T}, \text{TKeyExists}

Method

Checks if the entity exists for given predicate.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
<th>C++</th>
<th>F#</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>bool Exists(Expression&lt;Func&lt;T, bool&gt;&gt; predicate)</code></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Parameters

**predicate**
- Type: `System.Linq.Expressions.Expression<Func{T, Boolean}>`  
The expression.

### Return Value
- Type: `Boolean`  
True when an entity matching the predicate exists, false otherwise.

### See Also

**Reference**
- `IRepository{T}, TKey Interface`  
- `MongoRepository Namespace`
Send comments on this topic to Rob Janssen
IRepository\(T\), TKey\(\)\(\)GetById Method

Returns the \(T\) by its given id.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

```csharp
T GetById(TKey id)
```

### Parameters

**id**
Type: TKey  
The value representing the ObjectId of the entity to retrieve.

### Return Value
Type: \(T\)
The Entity \(T\).

### See Also

Reference  
IRepository\(T\), TKey Interface  
MongoRepository Namespace
Send comments on this topic to Rob Janssen
IRepository<T, TKey>RequestDone
Method

Lets the server know that this thread is done with a series of related operations.

Namespace: MongoRepository
Version: 1.6.11.0 (1.6.11.0)

Syntax

```csharp
void RequestDone()
```

Remarks

Instead of calling this method it is better to put the return value of RequestStart in a using statement.

See Also

Reference

IRepository<T, TKey> Interface
MongoRepository Namespace
**IRepository**{\textit{T, TKey}}\texttt{RequestStart}\ Method

Lets the server know that this thread is about to begin a series of related operations that must all occur on the same connection. The return value of this method implements IDisposable and can be placed in a using statement (in which case RequestDone will be called automatically when leaving the using statement).

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

```csharp
IDisposable RequestStart()
```

### Return Value

**Type:** IDisposable  
A helper object that implements IDisposable and calls RequestDone() from the Dispose method.

### Remarks

Sometimes a series of operations needs to be performed on the same connection in order to guarantee correct results. This is rarely the case, and most of the time there is no need to call RequestStart/RequestDone. An example of when this might be necessary is when a series of Inserts are called in rapid succession with SafeMode off, and you want to query that data in a consistent manner immediately thereafter (with SafeMode off the writes can queue up at the server and might not be immediately visible to other
connections). Using RequestStart you can force a query to be on the same connection as the writes, so the query won't execute until the server has caught up with the writes. A thread can temporarily reserve a connection from the connection pool by using RequestStart and RequestDone. You are free to use any other databases as well during the request. RequestStart increments a counter (for this thread) and RequestDone decrements the counter. The connection that was reserved is not actually returned to the connection pool until the count reaches zero again. This means that calls to RequestStart/RequestDone can be nested and the right thing will happen.

See Also

Reference
IRepositoryT, TKey Interface
MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
**IRepository{T, TKey} Update Method**

**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Update(IEnumerable{T})</td>
<td>Upserts the entities.</td>
</tr>
<tr>
<td>Update(T)</td>
<td>Upserts an entity.</td>
</tr>
</tbody>
</table>

**See Also**

Reference

IRepository{T, TKey} Interface

MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
IEnumerable\text{T, TKey}&nbsp;\text{Update}\&nbsp;Method\&nbsp;(&nbsp;IEnumerable\text{T}&nbsp;)

Upserts the entities.

\textbf{Namespace: }MongoRepository  \\
Version: 1.6.11.0 (1.6.11.0)

\section*{Syntax}

\begin{verbatim}
void Update(  
    IEnumerable\text{&lt;T&gt;} entities
)
\end{verbatim}

Parameters

\textit{entities}

Type: System.Collections.Generic\text{IEnumerable}\text{T}

The entities to update.

\section*{See Also}

Reference

IRepository\text{T, TKey} Interface  \\
Update Overload  \\
MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
IRepository\textit{T}, \textit{TKey}\texttt{Update} Method (\textit{T})

Upserts an entity.

\textbf{Namespace:} MongoRepository  
Version: 1.6.11.0 (1.6.11.0)

\section*{Syntax}

\begin{verbatim}
T Update(
    T entity
)
\end{verbatim}

\section*{Parameters}

\textit{entity}  
Type: \textit{T}  
The entity.

\section*{Return Value}  
Type: \textit{T}  
The updated entity.

\section*{See Also}

Reference  
IRepository\textit{T}, \textit{TKey} Interface  
Update Overload  
MongoRepository Namespace
IRepositoryManager<T> Interface

IRepositoryManager definition.

Namespace: MongoRepository
Version: 1.6.11.0 (1.6.11.0)

Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
<th>C++</th>
<th>F#</th>
</tr>
</thead>
<tbody>
<tr>
<td>public interface IRepositoryManager&lt;T&gt; : IRepositoryManager&lt;T, TKey&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Type Parameters

T

The type contained in the repository to manage.

The IRepositoryManager<T> type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exists</td>
<td>Gets a value indicating whether the collection already exists. (Inherited from IRepositoryManager&lt;T, TKey&gt;)</td>
</tr>
<tr>
<td>Name</td>
<td>Gets the name of the collection as Mongo uses. (Inherited from IRepositoryManager&lt;T, TKey&gt;)</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drop</td>
<td>Drops the repository. (Inherited from IRepositoryManagerT, TKey.)</td>
</tr>
<tr>
<td>DropAllIndexes</td>
<td>Drops all indexes on this repository. (Inherited from IRepositoryManagerT, TKey.)</td>
</tr>
<tr>
<td>DropIndex</td>
<td>Drops specified index on the repository. (Inherited from IRepositoryManagerT, TKey.)</td>
</tr>
<tr>
<td>DropIndexes</td>
<td>Drops specified indexes on the repository. (Inherited from IRepositoryManagerT, TKey.)</td>
</tr>
<tr>
<td>EnsureIndex(String)</td>
<td>Ensures that the desired index exist and creates it if it doesn't exist. (Inherited from IRepositoryManagerT, TKey.)</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>EnsureIndex(String, Boolean, Boolean, Boolean)</code></td>
<td>Ensures that the desired index exist and creates it if it doesn't exist. (Inherited from <code>IRepositoryManagerT, TKey</code>.)</td>
</tr>
<tr>
<td><code>EnsureIndexes(IEnumerable&lt;String&gt;)</code></td>
<td>Ensures that the desired indexes exist and creates them if they don't exist. (Inherited from <code>IRepositoryManagerT, TKey</code>.)</td>
</tr>
<tr>
<td><code>EnsureIndexes(IMongoIndexKeys, IMongoIndexOptions)</code></td>
<td>Ensures that the desired indexes exist and creates them if they don't exist. (Inherited from <code>IRepositoryManagerT, TKey</code>.)</td>
</tr>
<tr>
<td><code>EnsureIndexes(IEnumerable&lt;String&gt;, Boolean, Boolean, Boolean)</code></td>
<td>Ensures that the desired indexes exist and creates them if they don't exist. (Inherited from <code>IRepositoryManagerT, TKey</code>.)</td>
</tr>
<tr>
<td><code>GetIndexes</code></td>
<td>Gets the indexes for this repository. (Inherited from <code>IRepositoryManagerT, TKey</code>.)</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>GetStats</td>
<td>Gets stats for this repository. (Inherited from IRepositoryManager(T, TKey).)</td>
</tr>
<tr>
<td>GetTotalDataSize</td>
<td>Gets the total size for the repository (data + indexes). (Inherited from IRepositoryManager(T, TKey).)</td>
</tr>
<tr>
<td>GetTotalStorageSize</td>
<td>Gets the total storage size for the repository (data + indexes). (Inherited from IRepositoryManager(T, TKey).)</td>
</tr>
<tr>
<td>IndexesExists</td>
<td>Tests whether indexes exist. (Inherited from IRepositoryManager(T, TKey).)</td>
</tr>
<tr>
<td>IndexExists</td>
<td>Tests whether indexes exist. (Inherited from IRepositoryManager(T, TKey).)</td>
</tr>
<tr>
<td>IsCapped</td>
<td>Tests whether the repository is capped. (Inherited from IRepositoryManager(T, TKey).)</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>--------------------------------------------</td>
</tr>
<tr>
<td>ReIndex</td>
<td>Runs the ReIndex command on this repository. (Inherited from IRepositoryManagerT, TKey.)</td>
</tr>
<tr>
<td>Validate</td>
<td>Validates the integrity of the repository. (Inherited from IRepositoryManagerT, TKey.)</td>
</tr>
</tbody>
</table>

**Remarks**

Entities are assumed to use strings for Id's.

**See Also**

Reference

MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
IRepositoryManager<T> Properties

The IRepositoryManager<T> generic type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exists</td>
<td>Gets a value indicating whether the collection already exists. (Inherited from IRepositoryManager&lt;T, TKey.)</td>
</tr>
<tr>
<td>Name</td>
<td>Gets the name of the collection as Mongo uses. (Inherited from IRepositoryManager&lt;T, TKey.)</td>
</tr>
</tbody>
</table>

See Also

Reference
IRepositoryManager<T> Interface
MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
The **IRepositoryManagerT** generic type exposes the following members.

## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drop</td>
<td>Drops the repository. (Inherited from <strong>IRepositoryManagerT</strong>, <strong>TKey</strong>.)</td>
</tr>
<tr>
<td>DropAllIndexes</td>
<td>Drops all indexes on this repository. (Inherited from <strong>IRepositoryManagerT</strong>, <strong>TKey</strong>.)</td>
</tr>
<tr>
<td>DropIndex</td>
<td>Drops specified index on the repository. (Inherited from <strong>IRepositoryManagerT</strong>, <strong>TKey</strong>.)</td>
</tr>
<tr>
<td>DropIndexes</td>
<td>Drops specified indexes on the repository. (Inherited from <strong>IRepositoryManagerT</strong>, <strong>TKey</strong>.)</td>
</tr>
<tr>
<td>EnsureIndex(String)</td>
<td>Ensures that the desired index exist</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>EnsureIndex(String, Boolean, Boolean, Boolean)</td>
<td>Ensures that the desired index exist and creates it if it doesn't exist. The desired index exist and creates it if it doesn't exist. (Inherited from IRepositoryManager{T, TKey}.)</td>
</tr>
<tr>
<td>EnsureIndexes(IEnumerable&lt;String&gt;)</td>
<td>Ensures that the desired indexes exist and creates them if they don't exist.</td>
</tr>
<tr>
<td>EnsureIndexes(IMongoIndexKeys, IMongoIndexOptions)</td>
<td>Ensures that the desired indexes exist and creates them if they don't exist.</td>
</tr>
<tr>
<td>EnsureIndexes(IEnumerable&lt;String&gt;, Boolean, Boolean, Boolean)</td>
<td>Ensures that the desired indexes exist and creates them if they don't exist.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>GetIndexes</td>
<td>Gets the indexes for this repository.</td>
</tr>
<tr>
<td>GetStats</td>
<td>Gets stats for this repository.</td>
</tr>
<tr>
<td>GetTotalDataSize</td>
<td>Gets the total size for the repository (data + indexes).</td>
</tr>
<tr>
<td>GetTotalStorageSize</td>
<td>Gets the total storage size for the repository (data + indexes).</td>
</tr>
<tr>
<td>IndexesExists</td>
<td>Tests whether indexes exist.</td>
</tr>
<tr>
<td>IndexExists</td>
<td>Tests whether indexes exist.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>IsCapped</td>
<td>Tests whether the repository is capped. (Inherited from IRepositoryManagerT, TKey.)</td>
</tr>
<tr>
<td>ReIndex</td>
<td>Runs the ReIndex command on this repository. (Inherited from IRepositoryManagerT, TKey.)</td>
</tr>
<tr>
<td>Validate</td>
<td>Validates the integrity of the repository. (Inherited from IRepositoryManagerT, TKey.)</td>
</tr>
</tbody>
</table>

### See Also

**Reference**
- IRepositoryManagerT Interface
- MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
**IRepositoryManager<T, TKey> Interface**

IRepositoryManager definition.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

## Syntax

```csharp
public interface IRepositoryManager<T, TKey>
where T : Object, IEntity<TKey>
```

### Type Parameters

- **T**  
  The type contained in the repository to manage.

- **TKey**  
  The type used for the entity's Id.

The **IRepositoryManager<T, TKey>** type exposes the following members.

## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exists</td>
<td>Gets a value indicating whether the collection already exists.</td>
</tr>
<tr>
<td>Name</td>
<td>Gets the name of the collection as Mongo uses.</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drop</td>
<td>Drops the repository.</td>
</tr>
<tr>
<td>DropAllIndexes</td>
<td>Drops all indexes on this repository.</td>
</tr>
<tr>
<td>DropIndex</td>
<td>Drops specified index on the repository.</td>
</tr>
<tr>
<td>DropIndexes</td>
<td>Drops specified indexes on the repository.</td>
</tr>
<tr>
<td>EnsureIndex(String)</td>
<td>Ensures that the desired index exist and creates it if it doesn't exist.</td>
</tr>
<tr>
<td>EnsureIndex(String, Boolean, Boolean, Boolean)</td>
<td>Ensures that the desired index exist and creates it if it doesn't exist.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>EnsureIndexes(IEnumerable&lt;String&gt;)</td>
<td>Ensures that the desired indexes exist and creates them if they don't exist.</td>
</tr>
<tr>
<td>EnsureIndexes(IMongoIndexKeys, IMongoIndexOptions)</td>
<td>Ensures that the desired indexes exist and creates them if they don't exist.</td>
</tr>
<tr>
<td>EnsureIndexes(IEnumerable&lt;String&gt;, Boolean, Boolean, Boolean)</td>
<td>Ensures that the desired indexes exist and creates them if they don't exist.</td>
</tr>
<tr>
<td>GetIndexes</td>
<td>Gets the indexes for this repository.</td>
</tr>
<tr>
<td>GetStats</td>
<td>Gets stats for this repository.</td>
</tr>
<tr>
<td>GetTotalDataSize</td>
<td>Gets the total size for the repository (data + indexes).</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>GetTotalStorageSize</td>
<td>Gets the total storage size for the repository (data + indexes).</td>
</tr>
<tr>
<td>IndexesExists</td>
<td>Tests whether indexes exist.</td>
</tr>
<tr>
<td>IndexExists</td>
<td>Tests whether indexes exist.</td>
</tr>
<tr>
<td>IsCapped</td>
<td>Tests whether the repository is capped.</td>
</tr>
<tr>
<td>ReIndex</td>
<td>Runs the ReIndex command on this repository.</td>
</tr>
<tr>
<td>Validate</td>
<td>Validates the integrity of the repository.</td>
</tr>
</tbody>
</table>
The `IRepositoryManager<T, TKey>` generic type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exists</td>
<td>Gets a value indicating whether the collection already exists.</td>
</tr>
<tr>
<td>Name</td>
<td>Gets the name of the collection as Mongo uses.</td>
</tr>
</tbody>
</table>

### See Also

- **Reference**
  - `IRepositoryManager<T, TKey>` Interface
  - `MongoRepository` Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
IRepositoryManager{T, TKey} Exists Property

Gets a value indicating whether the collection already exists.

**Namespace:** MongoDBRepository  
**Assembly:** MongoDBRepository.Net45 (in MongoDBRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

```csharp
bool Exists { get; }
```

#### Property Value

Type: **Boolean**  
Returns true when the collection already exists, false otherwise.

### See Also

Reference  
IRepositoryManager{T, TKey} Interface  
MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
**IRepositoryManager{T, TKey} Name Property**

Gets the name of the collection as Mongo uses.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

```
string Name { get; }
```

### Property Value

**Type:** String  
The name of the collection as Mongo uses.

### See Also

**Reference**  
IRepositoryManager{T, TKey} Interface  
MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
The `IRepositoryManager<T, TKey>` generic type exposes the following members.

## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drop</td>
<td>Drops the repository.</td>
</tr>
<tr>
<td>DropAllIndexes</td>
<td>Drops all indexes on this repository.</td>
</tr>
<tr>
<td>DropIndex</td>
<td>Drops specified index on the repository.</td>
</tr>
<tr>
<td>DropIndexes</td>
<td>Drops specified indexes on the repository.</td>
</tr>
<tr>
<td>EnsureIndex(String)</td>
<td>Ensures that the desired index exist and creates it if it doesn't.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td><code>EnsureIndex(String, Boolean, Boolean, Boolean)</code></td>
<td>Ensures that the desired index exist and creates it if it doesn't exist.</td>
</tr>
<tr>
<td><code>EnsureIndexes(IEnumerable&lt;String&gt;)</code></td>
<td>Ensures that the desired indexes exist and creates them if they don't exist.</td>
</tr>
<tr>
<td><code>EnsureIndexes(IMongoIndexKeys, IMongoIndexOptions)</code></td>
<td>Ensures that the desired indexes exist and creates them if they don't exist.</td>
</tr>
<tr>
<td><code>EnsureIndexes(IEnumerable&lt;String&gt;, Boolean, Boolean, Boolean)</code></td>
<td>Ensures that the desired indexes exist and creates them if they don't exist.</td>
</tr>
<tr>
<td><code>GetIndexes</code></td>
<td>Gets the indexes for this repository.</td>
</tr>
<tr>
<td><code>GetStats</code></td>
<td>Gets stats for this repository.</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>GetTotalDataSize</td>
<td>Gets the total size for the repository (data + indexes).</td>
</tr>
<tr>
<td>GetTotalStorageSize</td>
<td>Gets the total storage size for the repository (data + indexes).</td>
</tr>
<tr>
<td>IndexesExists</td>
<td>Tests whether indexes exist.</td>
</tr>
<tr>
<td>IndexExists</td>
<td>Tests whether indexes exist.</td>
</tr>
<tr>
<td>IsCapped</td>
<td>Tests whether the repository is capped.</td>
</tr>
<tr>
<td>ReIndex</td>
<td>Runs the ReIndex command on this repository.</td>
</tr>
<tr>
<td>Validate</td>
<td>Validates the integrity of</td>
</tr>
</tbody>
</table>
See Also

Reference
- `IRepositoryManager{T, TKey Interface}
- `MongoRepository` Namespace
IRepositoryManager<T, TKey> Drop Method

Drops the repository.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

**C#**
```csharp
void Drop()
```

**VB**
```vbnet
Dim Drop As Void
```

**C++**
```cpp
void Drop();
```

**F#**
```fsharp
let Drop () =
```

### See Also

Reference  
IRepositoryManager<T, TKey> Interface  
MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl  
Send comments on this topic to Rob Janssen
IRepositoryManager\text{T}, TKeyDropAllIndexes Method

Drops all indexes on this repository.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

```csharp
void DropAllIndexes()
```

### See Also

Reference

IRepositoryManager\text{T}, TKey Interface  
MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
IRepositoryManager<T>, TKey DropIndex Method

Drops specified index on the repository.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
<th>C++</th>
<th>F#</th>
</tr>
</thead>
</table>

```csharp
void DropIndex(
    string keyname
)
```

### Parameters

**keyname**

Type: **System.String**

The name of the indexed field.

### See Also

Reference

IRepositoryManager<T, TKey> Interface  
MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
IRepositoryManagerT, TKeyDropIndexes Method

Drops specified indexes on the repository.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

```csharp
void DropIndexes(
    IEnumerable<string> keynames
)
```

### Parameters

- **keynames**
  - Type: `System.Collections.Generic.IEnumerable<string>`
  - The names of the indexed fields.

### See Also

**Reference**

- IRepositoryManagerT, TKey Interface  
- MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
IRepositoryManager<sup>T</sup>, TKey

EnsureIndex Method

## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![exclamation] EnsureIndex(String)</td>
<td>Ensures that the desired index exist and creates it if it doesn't exist.</td>
</tr>
<tr>
<td>![exclamation] EnsureIndex(String, Boolean, Boolean, Boolean)</td>
<td>Ensures that the desired index exist and creates it if it doesn't exist.</td>
</tr>
</tbody>
</table>

## See Also

Reference

IRepositoryManager<sup>T</sup>, TKey Interface
MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
IRepositoryManager<T, TKey>EnsureIndex Method (String)

Ensures that the desired index exist and creates it if it doesn't exist.

Namespace: MongoRepository
Version: 1.6.11.0 (1.6.11.0)

Syntax

```csharp
void EnsureIndex(
    string keyname
)
```

Parameters

- `keyname`
  - Type: System.String
  - The indexed field.

Remarks

This is a convenience method for EnsureIndexes(IMongoIndexKeys keys, IMongoIndexOptions options). Index will be ascending order, non-unique, non-sparse.

See Also

Reference
IRepositoryManagerT, TKey Interface
EnsureIndex Overload
MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
EnsureIndex method

Ensures that the desired index exist and creates it if it doesn't exist.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
<th>C++</th>
<th>F#</th>
</tr>
</thead>
</table>

```csharp
void EnsureIndex(
    string keyname,
    bool descending,
    bool unique,
    bool sparse
)
```

### Parameters

- **keyname**
  - Type: `System.String`
  - The indexed field.
- **descending**
  - Type: `System.Boolean`
  - Set to true to make index descending, false for ascending.
- **unique**
  - Type: `System.Boolean`
Set to true to ensure index enforces unique values.

\emph{sparse}

Type: \texttt{System.Boolean}

Set to true to specify the index is sparse.

\section*{Remarks}

This is a convenience method for \texttt{EnsureIndexes(IMongoIndexKeys
keys, IMongoIndexOptions options)}.

\section*{See Also}

Reference

\texttt{IRepositoryManager\textbackslash T, TKey Interface}

\texttt{EnsureIndex Overload}

\texttt{MongoRepository Namespace}

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
# IRepositoryManager<T, TKey> EnsureIndexes Method

## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EnsureIndexes(IEnumerable&lt;string&gt;)</td>
<td>Ensures that the desired indexes exist and creates them if they don't exist.</td>
</tr>
<tr>
<td>EnsureIndexes(IMongoIndexKeys, IMongoIndexOptions)</td>
<td>Ensures that the desired indexes exist and creates them if they don't exist.</td>
</tr>
<tr>
<td>EnsureIndexes(IEnumerable&lt;string&gt;, Boolean, Boolean, Boolean)</td>
<td>Ensures that the desired indexes exist and creates them if they don't exist.</td>
</tr>
</tbody>
</table>

## See Also

Reference
EnsureIndexes Method (IEnumerable<String>)

Ensures that the desired indexes exist and creates them if they don't exist.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
<th>C++</th>
<th>F#</th>
</tr>
</thead>
</table>

```csharp
void EnsureIndexes(  
    IEnumerable<string> keynames  
)
```

### Parameters

- `keynames`  
  Type: System.Collections.Generic(IEnumerable<String>)  
  The indexed fields.

### Remarks

This is a convenience method for EnsureIndexes(IMongoIndexKeys keys, IMongoIndexOptions options). Index will be ascending order, non-unique, non-sparse.

### See Also
Reference
IRepositoryManagerT, TKey Interface
EnsureIndexes Overload
MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
IRepositoryManager<T>, TKey

EnsureIndexes Method (IMongoIndexKeys, IMongoIndexOptions)

Ensures that the desired indexes exist and creates them if they don't exist.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

## Syntax

```csharp
void EnsureIndexes(
    IMongoIndexKeys keys,
    IMongoIndexOptions options
)
```

### Parameters

**keys**
- Type: IMongoIndexKeys
  - The indexed fields.

**options**
- Type: IMongoIndexOptions
  - The index options.

## Remarks
This is a convenience method for EnsureIndexes(IMongoIndexKeys keys, IMongoIndexOptions options). Index will be ascending order, non-unique, non-sparse.

See Also

Reference

IRepositoryManagerT, TKey Interface
EnsureIndexes Overload
MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
IRepositoryManager\text{T}, 
\text{TKey}\text{EnsureIndexes Method (IEnumerable\text{String}, Boolean, Boolean, Boolean)}

Ensures that the desired indexes exist and creates them if they don't exist.

\textbf{Namespace:} MongoRepository  
\textbf{Version:} 1.6.11.0 (1.6.11.0)

\section*{Syntax}

\begin{verbatim}
void EnsureIndexes(
    IEnumerable<string> keynames,
    bool descending,
    bool unique,
    bool sparse
)
\end{verbatim}

\section*{Parameters}

\subsubsection*{keynames}
Type: \textit{System.Collections.Generic\{IEnumerable\text{String}}

The indexed fields.

\subsubsection*{descending}
Type: \textit{System\{Boolean}}

Set to true to make index descending, false for ascending.

\subsubsection*{unique}
Type: **SystemBoolean**
Set to true to ensure index enforces unique values.

*sparse*
Type: **SystemBoolean**
Set to true to specify the index is sparse.

**Remarks**
This is a convenience method for EnsureIndexes(IMongoIndexKeys keys, IMongoIndexOptions options).

**See Also**
Reference
- IRepositoryManagerT, TKey Interface
- EnsureIndexes Overload
- MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl
Send comments on this topic to Rob Janssen
IRepositoryManager\(T, TKey\) GetIndexes Method

Gets the indexes for this repository.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
<th>C++</th>
<th>F#</th>
<th>Copy</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetIndexesResult GetIndexes()</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Return Value**

Type: `GetIndexesResult`  
Returns the indexes for this repository.

### See Also

Reference

IRepositoryManager\(T, TKey\) Interface  
MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl  
Send comments on this topic to Rob Janssen
IRepositoryManager$T, TKey GetStats Method

Gets stats for this repository.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
<th>C++</th>
<th>F#</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```csharp
CollectionStatsResult GetStats()
```

Return Value  
Type: **CollectionStatsResult**  
Returns a CollectionStatsResult.

### Remarks

You will need to reference MongoDB.Driver.

### See Also

Reference  
IRepositoryManager$T, TKey Interface  
MongoRepository Namespace
IRepositoryManager<T, TKey> GetTotalDataSize Method

Gets the total size for the repository (data + indexes).

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

## Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
<th>C++</th>
<th>F#</th>
</tr>
</thead>
<tbody>
<tr>
<td>long GetTotalDataSize()</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Return Value

Type: `Int64`

Returns total size for the repository (data + indexes).

## See Also

Reference

IRepositoryManager<T, TKey> Interface  
MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
IRepositoryManager\text{T,} T Key

\text{GetTotalStorageSize Method}

Gets the total storage size for the repository (data + indexes).

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
<th>C++</th>
<th>F#</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

    long GetTotalStorageSize()

**Return Value**

Type: **Int64**  
Returns total storage size for the repository (data + indexes).

### See Also

Reference

- IRepositoryManager\text{T,} T Key Interface  
- MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
IRepositoryManager\(T\), TKeyIndexesExists Method

Tests whether indexes exist.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
<th>C++</th>
<th>F#</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```csharp
bool IndexesExists(
    IEnumerable<string> keynames
)
```

### Parameters

- **keynames**
  - Type: `System.Collections.Generic(IEnumerable<string>`  
  - The indexed fields.

### Return Value

- **Type:** Boolean  
- Returns true when the indexes exist, false otherwise.

### See Also

- **Reference**  
  - IRepositoryManager\(T\), TKey Interface  
  - MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl
Send comments on this topic to Rob Janssen
IRepositoryManager<T, TKey> IndexExists Method

Tests whether indexes exist.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

```csharp
bool IndexExists(
    string keyname
)
```

### Parameters

`keyname`  
Type: `System.String`  
The indexed fields.

### Return Value

Type: `Boolean`  
Returns true when the indexes exist, false otherwise.

### See Also

**Reference**  
IRepositoryManager<T, TKey> Interface  
MongoRepository Namespace
Send comments on this topic to Rob Janssen
IRepositoryManagerT, TKey IsCapped Method

Tests whether the repository is capped.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
<th>C++</th>
<th>F#</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```csharp
bool IsCapped()
```

**Return Value**  
Type: **Boolean**  
Returns true when the repository is capped, false otherwise.

### See Also

Reference  
IRepositoryManagerT, TKey Interface  
MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
IRepositoryManager{T, TKey} ReIndex Method

Runs the ReIndex command on this repository.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

```csharp
void ReIndex()
```

### See Also

Reference  
IRepositoryManager{T, TKey} Interface  
MongoRepository Namespace
IRepositoryManager<T, TKey> Validate Method

Validates the integrity of the repository.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
<th>C++</th>
<th>F#</th>
<th>Copy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```csharp
ValidateCollectionResult Validate()
```

**Return Value**

**Type:** ValidateCollectionResult  
Returns a ValidateCollectionResult.

### Remarks

You will need to reference MongoDB.Driver.

### See Also

**Reference**  
IRepositoryManager<T, TKey> Interface  
MongoRepository Namespace
MongoRepositoryT Class

Deals with entities in MongoDB.

▲ Inheritance Hierarchy

```
System
   Object
  MongoRepository
```

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

▲ Syntax

```
public class MongoRepository<T> : MongoRepository
    IRepository<T>, IRepository<T, string>, IQueryable
where T : Object, IEntity<string>
```

**Type Parameters**

```
T
```

The type contained in the repository.

The `MongoRepositoryT` type exposes the following members.

▲ Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>MongoRepositoryT</code></td>
<td>Initializes a new instance of the</td>
</tr>
</tbody>
</table>
MongoRepository class. Uses the Default App/Web.Config connectionstrings to fetch the connectionString and Database name.

- **MongoRepositoryT(MongoUrl)**: Initializes a new instance of the MongoRepository class.
- **MongoRepositoryT(String)**: Initializes a new instance of the MongoRepository class.
- **MongoRepositoryT(MongoUrl, String)**: Initializes a new instance of the MongoRepository class.
- **MongoRepositoryT(String, String)**: Initializes a new instance of the MongoRepository class.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collection</td>
<td>Gets the Mongo collection (to perform advanced operations).</td>
</tr>
</tbody>
</table>
(Inherited from MongoDBRepository{T, TKey}.)

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Add(IEnumerable{T})</strong></td>
<td>Adds the new entities in the repository. (Inherited from MongoDBRepository{T, TKey}.)</td>
</tr>
<tr>
<td>Action</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Add(T)</strong></td>
<td>Adds the new entity in the repository. (Inherited from MongoRepositoryT, TKey.)</td>
</tr>
<tr>
<td><strong>Count</strong></td>
<td>Counts the total entities in the repository. (Inherited from MongoRepositoryT, TKey.)</td>
</tr>
<tr>
<td><strong>Delete(ObjectId)</strong></td>
<td>Deletes an entity from the repository by its ObjectId. (Inherited from MongoRepositoryT, TKey.)</td>
</tr>
<tr>
<td><strong>Delete(ExpressionFuncT, Boolean)</strong></td>
<td>Deletes the entities matching the predicate. (Inherited from MongoRepositoryT, TKey.)</td>
</tr>
<tr>
<td><strong>Delete(TKey)</strong></td>
<td>Deletes an entity from the repository by its id. (Inherited from MongoRepositoryT, TKey.)</td>
</tr>
<tr>
<td><strong>DeleteAll</strong></td>
<td>Deletes all entities in the repository. (Inherited from MongoRepositoryT, TKey.)</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>Exists</strong></td>
<td>Checks if the entity exists for given predicate. (Inherited from <strong>MongoRepository</strong>&lt;T, TKey&gt;.)</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>GetById(ObjectId)</strong></td>
<td>Returns the T by its given id. (Inherited from <strong>MongoRepository</strong>&lt;T, TKey&gt;.)</td>
</tr>
<tr>
<td><strong>GetById(TKey)</strong></td>
<td>Returns the T by its given id. (Inherited from <strong>MongoRepository</strong>&lt;T, TKey&gt;.)</td>
</tr>
<tr>
<td><strong>GetEnumerator</strong></td>
<td>Returns an enumerator that iterates through a collection.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>RequestDone</td>
<td>Lets the server know that this thread is done with a series of related operations. (Inherited from MongoRepositoryT, TKey.)</td>
</tr>
<tr>
<td>RequestStart</td>
<td>Lets the server know that this thread is about to begin a series of related operations that must all occur on the same connection. The return value of this method implements IDisposable and can be placed in a using statement (in which case RequestDone will be called automatically</td>
</tr>
</tbody>
</table>
when leaving the using statement).
(Inherited from MongoRepository{T, TKey}.)

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Update(IEnumerable{T})</td>
<td>Upserts the entities. (Inherited from MongoRepository{T, TKey}.)</td>
</tr>
<tr>
<td>Update(T)</td>
<td>Upserts an entity. (Inherited from MongoRepository{T, TKey}.)</td>
</tr>
</tbody>
</table>

**Fields**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>collection</td>
<td>MongoCollection field. (Inherited from MongoRepository{T, TKey}.)</td>
</tr>
</tbody>
</table>

**Remarks**

Entities are assumed to use strings for Id's.
See Also

Reference

MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
# MongoRepository<sup>T</sup> Constructor

## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MongoRepository&lt;sup&gt;T&lt;/sup&gt;</td>
<td>Initializes a new instance of the MongoRepository class. Uses the Default App/Web.Config connectionstrings to fetch the connectionString and Database name.</td>
</tr>
<tr>
<td>MongoRepository&lt;sup&gt;T&lt;/sup&gt;(MongoUrl)</td>
<td>Initializes a new instance of the MongoRepository class.</td>
</tr>
<tr>
<td>MongoRepository&lt;sup&gt;T&lt;/sup&gt;(String)</td>
<td>Initializes a new instance of the MongoRepository class.</td>
</tr>
<tr>
<td>MongoRepository&lt;sup&gt;T&lt;/sup&gt;(MongoUrl, String)</td>
<td>Initializes a new instance of the MongoRepository class.</td>
</tr>
<tr>
<td>MongoRepository&lt;sup&gt;T&lt;/sup&gt;(String,</td>
<td></td>
</tr>
</tbody>
</table>

...
String) instance of the MongoRepository class.

See Also

Reference

MongoRepositoryT Class
MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
MongoRepositoryT Constructor

Initializes a new instance of the MongoRepository class. Uses the Default App/Web.Config connectionstrings to fetch the connectionString and Database name.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

```c#
public MongoRepository()
```

### Remarks

Default constructor defaults to "MongoServerSettings" key for connectionstring.

### See Also

Reference  
- MongoRepositoryT Class  
- MongoRepositoryT Overload  
- MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl  
Send comments on this topic to Rob Janssen
MongoRepository documentation

MongoRepository<	Constructor
(MongoUrl)

Initializes a new instance of the MongoRepository class.

**Namespace:** MongoRepository

**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)

**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
<th>C++</th>
<th>F#</th>
</tr>
</thead>
</table>
| **public** MongoRepository(<
|   MongoDBUrl **url**
| ) |

### Parameters

**url**

Type: **MongoUrl**

Url to use for connecting to MongoDB.

### See Also

**Reference**

MongoRepository<T Class

MongoRepository<T Overload

MongoRepository Namespace

© 2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
MongoRepository\$ Constructor

Initializes a new instance of the MongoRepository class.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

## Syntax

### C#

```csharp
public MongoRepository(
    string connectionString
)
```

### Parameters

*connectionString*

Type: System.String  
Connectionstring to use for connecting to MongoDB.

## See Also

**Reference**

- MongoRepository\$ Class  
- MongoRepository\$ Overload  
- MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
MongoRepository\tdocumentation

MongoRepository\t\tT Constructor
(MongoUrl, String)

Initializes a new instance of the MongoRepository class.

**Namespace:** MongoRepository

**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)

**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
<th>C++</th>
<th>F#</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```csharp
public MongoRepository(
    MongoUrl url,
    string collectionName
)
```

#### Parameters

**url**

Type: **MongoUrl**

Url to use for connecting to MongoDB.

**collectionName**

Type: **System.String**

The name of the collection to use.

### See Also

**Reference**

- MongoRepository\tT Class
- MongoRepository\tT Overload
- MongoRepository\tNamespace
MongoRepository\text{T} Constructor (String, String)

Initializes a new instance of the MongoRepository class.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

C#  
```csharp
public MongoRepository(
    string connectionString,
    string collectionName
)
```

### Parameters

- **connectionString**  
  **Type:** System.String  
  Connectionstring to use for connecting to MongoDB.

- **collectionName**  
  **Type:** System.String  
  The name of the collection to use.

### See Also

- Reference  
  - MongoRepository\text{T} Class  
  - MongoRepository\text{T} Overload  
  - MongoRepository Namespace
The `MongoRepository<T>` generic type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Collection</strong></td>
<td>Gets the Mongo collection (to perform advanced operations). (Inherited from <code>MongoRepository&lt;T, TKey&gt;</code>)</td>
</tr>
<tr>
<td><strong>CollectionName</strong></td>
<td>Gets the name of the collection (Inherited from <code>MongoRepository&lt;T, TKey&gt;</code>)</td>
</tr>
<tr>
<td><strong>ElementType</strong></td>
<td>Gets the type of the element(s) that are returned when the expression tree associated with this instance of <code>IQueryable</code> is executed. (Inherited from <code>MongoRepository&lt;T, TKey&gt;</code>)</td>
</tr>
<tr>
<td><strong>Expression</strong></td>
<td>Gets the expression tree that is associated with the instance of <code>IQueryable</code>. (Inherited from <code>MongoRepository&lt;T, TKey&gt;</code>)</td>
</tr>
<tr>
<td><strong>Provider</strong></td>
<td>Gets the query provider that is associated with this data source.</td>
</tr>
</tbody>
</table>
(Inherited from MongoRepositoryT, TKey.)

Top

See Also

Reference
MongoRepositoryT Class
MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
The **MongoRepository** generic type exposes the following members.

## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add(IEnumerable&lt;T&gt;)</td>
<td>Adds the new entities in the repository. (Inherited from <code>MongoRepository{T,TKey}.</code>)</td>
</tr>
<tr>
<td>Add(T)</td>
<td>Adds the new entity in the repository. (Inherited from <code>MongoRepository{T,TKey}.</code>)</td>
</tr>
<tr>
<td>Count</td>
<td>Counts the total entities in the repository. (Inherited from <code>MongoRepository{T,TKey}.</code>)</td>
</tr>
<tr>
<td>Delete(ObjectId)</td>
<td>Deletes an entity from the repository by its ObjectId. (Inherited from <code>MongoRepository{T,TKey}.</code>)</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>Delete(Expression Func T, Boolean)</td>
<td>Deletes the entities matching the predicate. (Inherited from MongoRepository T, TKey.)</td>
</tr>
<tr>
<td>Delete(TKey)</td>
<td>Deletes an entity from the repository by its id. (Inherited from MongoRepository T, TKey.)</td>
</tr>
<tr>
<td>DeleteAll</td>
<td>Deletes all entities in the repository. (Inherited from MongoRepository T, TKey.)</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Exists</td>
<td>Checks if the entity exists for given predicate. (Inherited from MongoRepository T, TKey.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>GetById(ObjectId)</td>
<td>Returns the T by its given id. (Inherited from MongoRepositoryT, TKey.)</td>
</tr>
<tr>
<td>GetById(TKey)</td>
<td>Returns the T by its given id. (Inherited from MongoRepositoryT, TKey.)</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through a collection. (Inherited from MongoRepositoryT, TKey.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>RequestDone</td>
<td>Lets the server know that this thread is done with a series of related operations.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>RequestStart</td>
<td>Lets the server know that this thread is about to begin a series of related operations that must all occur on the same connection. The return value of this method implements IDisposable and can be placed in a using statement (in which case RequestDone will be called automatically when leaving the using statement). (Inherited from MongoRepositoryT, TKey.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Update(IEnumerableT)</td>
<td>Upserts the entities. (Inherited from MongoRepositoryT, TKey.)</td>
</tr>
<tr>
<td>Update(T)</td>
<td>Upserts an entity. (Inherited from MongoRepositoryT, TKey.)</td>
</tr>
</tbody>
</table>
See Also

Reference
MongoRepositoryT Class
MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
MongoRepository documentation

MongoRepository\texttt{T} Fields

The \texttt{MongoRepository\texttt{T}} generic type exposes the following members.

Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>\texttt{collection}</td>
<td>MongoCollection field. (Inherited from \texttt{MongoRepository\texttt{T}, TKey}.)</td>
</tr>
</tbody>
</table>

See Also

Reference

- \texttt{MongoRepository\texttt{T} Class}
- \texttt{MongoRepository Namespace}

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
MongoRepositoryT, TKey Class

Deals with entities in MongoDB.

Inheritance Hierarchy

SystemObject  MongoRepositoryMongoRepositoryT, TKey
             MongoRepositoryMongoRepositoryT

Namespace: MongoRepository
Version: 1.6.11.0 (1.6.11.0)

Syntax

```
public class MongoRepository<T, TKey> : IRepository<T>, IQueryable<T>, IEnumerable<T>, IQueryable
where T : Object, IEntity<TKey>
```

Type Parameters

T
The type contained in the repository.

TKey
The type used for the entity's Id.

The MongoRepositoryT, TKey type exposes the following members.

Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**MongoRepositoryT, TKey**
Initializes a new instance of the MongoRepository class. Uses the Default App/Web.Config connectionstrings to fetch the connectionString and Database name.

**Top**

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Collection</strong></td>
<td>Gets the Mongo collection (to perform advanced operations).</td>
</tr>
<tr>
<td><strong>CollectionName</strong></td>
<td>Gets the name of the collection</td>
</tr>
<tr>
<td><strong>ElementType</strong></td>
<td>Gets the type of the element(s) that are returned when the expression tree associated with this instance of IQueryable is executed.</td>
</tr>
</tbody>
</table>
Expression

Gets the expression tree that is associated with the instance of IQueryable.

Provider

Gets the query provider that is associated with this data source.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add(IEnumerable{T})</td>
<td>Adds the new entities in the repository.</td>
</tr>
<tr>
<td>Add(T)</td>
<td>Adds the new entity in the repository.</td>
</tr>
<tr>
<td>Count</td>
<td>Counts the total entities in the repository.</td>
</tr>
<tr>
<td>Delete(ObjectId)</td>
<td>Deletes an entity from the repository by its ObjectId.</td>
</tr>
<tr>
<td>Delete(ExpressionFuncT, Boolean)</td>
<td>Deletes the entities matching the predicate.</td>
</tr>
<tr>
<td>Delete(T)</td>
<td>Deletes the given entity.</td>
</tr>
<tr>
<td>Delete(TKey)</td>
<td>Deletes an entity from the repository by its id.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>DeleteAll</td>
<td>Deletes all entities in the repository.</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Exists</td>
<td>Checks if the entity exists for given predicate.</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetById(ObjectId)</td>
<td>Returns the T by its given id.</td>
</tr>
<tr>
<td>GetById(TKey)</td>
<td>Returns the T by its given id.</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through a collection.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a> (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td>RequestDone</td>
<td>Lets the server know that this thread is done with a series of related operations.</td>
</tr>
<tr>
<td>RequestStart</td>
<td>Lets the server know that this thread is about to begin a series of related operations that must all occur on the same connection. The return value of this method implements IDisposable and can be placed in a using statement (in which case RequestDone will be called automatically when leaving the using statement).</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td>Update(IEnumerable&lt;T&gt;)</td>
<td>Upserts the entities.</td>
</tr>
<tr>
<td>Update(T)</td>
<td>Upserts an entity.</td>
</tr>
</tbody>
</table>
 Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>collection</td>
<td>MongoCollection field.</td>
</tr>
</tbody>
</table>

See Also

Reference
MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
## Constructor

### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MongoRepository{T, TKey}</strong></td>
<td>Initializes a new instance of the MongoRepository class. Uses the Default App/Web.Config connectionstrings to fetch the connectionString and Database name.</td>
</tr>
<tr>
<td><strong>MongoRepository{T, TKey(MongoUrl)}</strong></td>
<td>Initializes a new instance of the MongoRepository class.</td>
</tr>
<tr>
<td><strong>MongoRepository{T, TKey(String)}</strong></td>
<td>Initializes a new instance of the MongoRepository class.</td>
</tr>
<tr>
<td><strong>MongoRepository{T, TKey(MongoUrl, String)}</strong></td>
<td>Initializes a new instance of the MongoRepository class.</td>
</tr>
<tr>
<td><strong>MongoRepository{T, TKey(String, String)}</strong></td>
<td>Initializes a new instance of the MongoRepository class.</td>
</tr>
</tbody>
</table>

### See Also
MongoRepository documentation

MongoRepository< T, TKey > Constructor

Initializes a new instance of the MongoRepository class. Uses the Default App/Web.Config connectionstrings to fetch the connectionString and Database name.

**Namespace**: MongoRepository  
**Version**: 1.6.11.0 (1.6.11.0)

### Syntax

```csharp
public MongoRepository()
```

### Remarks

Default constructor defaults to "MongoServerSettings" key for connectionString.

### See Also

Reference  
- [MongoRepository< T, TKey > Class](#)  
- [MongoRepository< T, TKey > Overload](#)  
- [MongoRepository Namespace](#)

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
MongoRepository documentation

MongoRepository T, TKey Constructor (MongoUrl)

Initializes a new instance of the MongoRepository class.

**Namespace:** MongoRepository

**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)

**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

```csharp
public MongoRepository(
    MongoUrl url
)
```

### Parameters

**url**

*Type: MongoUrl*

Url to use for connecting to MongoDB.

### See Also

Reference

MongoRepositoryT, TKey Class

MongoRepositoryT, TKey Overload

MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
MongoRepository\texttt{T}, \texttt{TKey} Constructor (String)

Initializes a new instance of the MongoRepository class.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
<th>C++</th>
<th>F#</th>
</tr>
</thead>
</table>
| ```csharp
public MongoRepository(
    string connectionString
)
``` |

### Parameters

- **connectionString**  
  Type: System.String  
  Connectionstring to use for connecting to MongoDB.

### See Also

**Reference**  
- MongoRepository\texttt{T}, \texttt{TKey} Class  
- MongoRepository\texttt{T}, \texttt{TKey} Overload  
- MongoRepository Namespace  

©2013 - 2015 Rob Janssen / Devcorner.nl  
Send comments on this topic to Rob Janssen
MongoRepository<T, TKey> Constructor (MongoUrl, String)

Initializes a new instance of the MongoRepository class.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

## Syntax

**C#**

```csharp
public MongoRepository(
    MongoUrl url,
    string collectionName
)
```

**See Also**

**Parameters**

*url*  
Type: **MongoUrl**  
Url to use for connecting to MongoDB.

*collectionName*  
Type: **SystemString**  
The name of the collection to use.

**Reference**

MongoRepository<T, TKey> Class  
MongoRepository<T, TKey> Overload  
MongoRepository Namespace
MongoRepository documentation

MongoRepository\textit{T, TKey} Constructor (String, String)

Initializes a new instance of the MongoRepository class.

**Namespace:** MongoRepository

**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)

**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

```csharp
public MongoRepository(
    string connectionString,
    string collectionName
)
```

### Parameters

- **connectionString**
  - Type: `System.String`
  - Connectionstring to use for connecting to MongoDB.

- **collectionName**
  - Type: `System.String`
  - The name of the collection to use.

### See Also

Reference

- MongoRepository\textit{T, TKey Class}
- MongoRepository\textit{T, TKey Overload}
- MongoRepository Namespace
MongoRepository\textit{T}, \textit{TKey} Properties

The \textit{MongoRepository\textit{T}, \textit{TKey}} generic type exposes the following members.

\textbf{Properties}

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collection</td>
<td>Gets the Mongo collection (to perform advanced operations).</td>
</tr>
<tr>
<td>CollectionName</td>
<td>Gets the name of the collection</td>
</tr>
<tr>
<td>ElementType</td>
<td>Gets the type of the element(s) that are returned when the expression tree associated with this instance of \textit{IQueryable} is executed.</td>
</tr>
<tr>
<td>Expression</td>
<td>Gets the expression tree that is associated with the instance of \textit{IQueryable}.</td>
</tr>
<tr>
<td>Provider</td>
<td>Gets the query provider that is associated with this data source.</td>
</tr>
</tbody>
</table>

\textbf{See Also}

Reference
MongoRepositoryT, TKey Class
MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
MongoRepository<T>, TKeyCollection Property

Gets the Mongo collection (to perform advanced operations).

Namespace: MongoRepository
Version: 1.6.11.0 (1.6.11.0)

Syntax

```csharp
public MongoCollection<T> Collection { get; }
```

Property Value

Type: `MongoCollection<T>`
The Mongo collection (to perform advanced operations).

Implements

`IRepository<T, TKeyCollection`

Remarks

One can argue that exposing this property (and with that, access to its Database property for instance (which is a "parent")) is not the responsibility of this class. Use of this property is highly discouraged; for most purposes you can use the MongoRepositoryManager<T>

See Also

Reference

MongoRepository<T, TKey Class
MongoRepository Namespace
MongoRepository Documentation

MongoRepositoryT, TKeyCollectionName Property

Gets the name of the collection

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
<th>C++</th>
<th>F#</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```csharp
public string CollectionName { get; }
```

**Property Value**  
**Type:** String

### See Also

**Reference**  
MongoRepositoryT, TKey Class  
MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
MongoRepository $T, TKey$ElementType $Property$

Gets the type of the element(s) that are returned when the expression tree associated with this instance of IQueryable is executed.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

```csharp
public virtual Type ElementType { get; }
```

### Property Value

Type: `Type`  
Implements  
IQueryable$ElementType$

### See Also

Reference  
MongoRepository$T, TKey$ Class  
MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
MongoRepository\text{T, TKey} Expression Property

Gets the expression tree that is associated with the instance of IQueryable.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

```csharp
public virtual Expression Expression { get; }
```

### Property Value

- **Type:** Expression
- Implements IQueryableExpression

### See Also

- Reference
  - MongoRepository\text{T, TKey} Class
  - MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
MongoRepository\T, TKeyProvider Property

Gets the query provider that is associated with this data source.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

**Syntax**

```csharp
public virtual IQueryProvider Provider { get; }
```

**Property Value**

Type: IQueryProvider

**Implements**

IQueryableProvider

**See Also**

**Reference**

MongoRepository\T, TKey Class  
MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
MongoRepository<T, TKey> Methods

The MongoRepository<T, TKey> generic type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add(IEnumerable&lt;T&gt;)</td>
<td>Adds the new entities in the repository.</td>
</tr>
<tr>
<td>Add(T)</td>
<td>Adds the new entity in the repository.</td>
</tr>
<tr>
<td>Count</td>
<td>Counts the total entities in the repository.</td>
</tr>
<tr>
<td>Delete(ObjectId)</td>
<td>Deletes an entity from the repository by its ObjectId.</td>
</tr>
<tr>
<td>Delete(Expression&lt;Func&lt;T, Boolean&gt;&gt;)</td>
<td>Deletes the entities matching the predicate.</td>
</tr>
<tr>
<td>Delete(T)</td>
<td>Deletes the given entity.</td>
</tr>
<tr>
<td>Delete(TKey)</td>
<td>Deletes an entity from the repository by its id.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DeleteAll</td>
<td>Deletes all entities in the repository.</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>Exists</td>
<td>Checks if the entity exists for given predicate.</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetByld(ObjectId)</td>
<td>Returns the T by its given id.</td>
</tr>
<tr>
<td>GetByld(TKey)</td>
<td>Returns the T by its given id.</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through a collection.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <code>Type</code> of the current instance.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object.</td>
</tr>
<tr>
<td>RequestDone</td>
<td>Lets the server know that this thread is done with a series of related operations.</td>
</tr>
<tr>
<td>RequestStart</td>
<td>Lets the server know that this thread is about to begin a series of related operations that must all occur on the same connection. The return value of this method implements IDisposable and can be placed in a using statement (in which case RequestDone will be called automatically when leaving the using statement).</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object.</td>
</tr>
<tr>
<td>Update(IEnumerable{T})</td>
<td>Upserts the entities.</td>
</tr>
<tr>
<td>Update(T)</td>
<td>Upserts an entity.</td>
</tr>
</tbody>
</table>
See Also

Reference
MongoRepositoryT, TKey Class
MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl
Send comments on this topic to Rob Janssen
# MongoRepository\textit{T}, TKey Add Method

## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add(IEnumerable\textit{T})</td>
<td>Adds the new entities in the repository.</td>
</tr>
<tr>
<td>Add(T)</td>
<td>Adds the new entity in the repository.</td>
</tr>
</tbody>
</table>

## See Also

Reference

- MongoRepository\textit{T}, TKey Class
- MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
MongoRepository \( T, \ TKey \) Add Method (IEnumerable \( T \))

Adds the new entities in the repository.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
<th>C++</th>
<th>F#</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```csharp
public virtual void Add(
    IEnumerable<T> entities
)
```

### Parameters

**entities**
- Type: `System.Collections.Generic(IEnumerable<T>`
- The entities of type \( T \).

### See Also

**Reference**  
MongoRepository \( T, \ TKey \) Class  
Add Overload  
MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl
Send comments on this topic to Rob Janssen
MongoRepository documentation

MongoRepository\(T, T\)\text{Key Add Method (}\(T)\text{)}

Adds the new entity in the repository.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
<th>C++</th>
<th>F#</th>
</tr>
</thead>
<tbody>
<tr>
<td>public virtual T Add(T entity)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Parameters

*entity*  
Type: \(T\)  
The entity \(T\).

Return Value  
Type: \(T\)  
The added entity including its new ObjectId.

Implements  
**IRepository\(T, T\)\text{Key Add(T)}**

### See Also

Reference  
MongoRepository\(T, T\)\text{Key Class Add Overload}
MongoRepository\(T, \text{TKey}\)Count Method

Counts the total entities in the repository.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

```
public virtual long Count()
```

**Return Value**

Type: **Int64**  
Count of entities in the collection.

**Implements**  
**IRepository\(T, \text{TKey}\)**

### See Also

**Reference**  
MongoRepository\(T, \text{TKey}\) Class  
MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
MongoRepository $T$, $TKey$ Delete Method

**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Icon] Delete(ObjectId)</td>
<td>Deletes an entity from the repository by its ObjectId.</td>
</tr>
<tr>
<td>![Icon] Delete(Expression.FuncT, Boolean)</td>
<td>Deletes the entities matching the predicate.</td>
</tr>
<tr>
<td>![Icon] Delete(T)</td>
<td>Deletes the given entity.</td>
</tr>
<tr>
<td>![Icon] Delete(TKey)</td>
<td>Deletes an entity from the repository by its id.</td>
</tr>
</tbody>
</table>

**See Also**

Reference

MongoRepository $T$, $TKey$ Class

MongoRepository Namespace

---

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
MongoRepository\textsuperscript{T}, TKey\texttt{Delete} Method (ObjectId)

Deletes an entity from the repository by its ObjectId.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

```csharp
public virtual void Delete(  
    ObjectId id
)
```

### Parameters

**id**

Type: **ObjectId**  
The ObjectId of the entity.

### See Also

Reference  
MongoRepository\textsuperscript{T}, TKey Class  
Delete Overload  
MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
MongoRepository documentation

MongoRepository<T, TKey>Delete Method (Expression<Func<T, Boolean>)

Deletes the entities matching the predicate.

**Namespace:** MongoRepository
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
<th>C++</th>
<th>F#</th>
</tr>
</thead>
</table>
| ```
public virtual void Delete(
    Expression<Func<T, bool>> predicate
)
``` |

#### Parameters

**predicate**

#### Implements
`IRepository<T, TKey>Delete(Expression<Func<T, Boolean>)`

### See Also

**Reference**
MongoRepository<T, TKey> Class
Delete Overload
MongoRepository Namespace
MongoRepository\textit{T}, \textit{TKey} Delete Method (\textit{T})

Deletes the given entity.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

```
public virtual void Delete(
    T entity
)
```

### Parameters

- **entity**
  - Type: \textit{T}
  - The entity to delete.

### Implements

\textit{IRepository\textit{T}, TKeyDelete(T)}

### See Also

- **Reference**
  - MongoRepository\textit{T}, TKey Class
  - Delete Overload
  - MongoRepository Namespace
Send comments on this topic to Rob Janssen
MongoRepository\textit{T}, \textit{TKey} \textbf{Delete Method (\textit{TKey})}

Deletes an entity from the repository by its id.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
<th>C++</th>
<th>F#</th>
</tr>
</thead>
</table>
| public virtual void Delete(  
  TKey \textit{id}  
) | | | |

**Parameters**

\textit{id}  
Type: \textit{TKey}  
The entity’s id.

**Implements**  
Repository\textit{T}, \textit{TKey}Delete\textit{TKey}•

### See Also

**Reference**  
MongoRepository\textit{T}, \textit{TKey} Class  
Delete Overload  
MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl
Send comments on this topic to Rob Janssen
MongoRepository\(T, TKey\)DeleteAll Method

Deletes all entities in the repository.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

```csharp
public virtual void DeleteAll()
```

### See Also

**Reference**  
MongoRepository\(T, TKey\) Class  
MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to [Rob Janssen](mailto:rob@devcorner.nl)
MongoRepository documentation

MongoRepository\texttt{T, TKey} Exists Method

Checks if the entity exists for given predicate.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
<th>C++</th>
<th>F#</th>
</tr>
</thead>
</table>
| public virtual bool Exists(
Expression<Func<T, bool>> \texttt{predicate}
) |

**Parameters**

\texttt{predicate}

Type: System.Linq.Expressions.Expression<Func\texttt{T, Boolean}}  
The expression.

**Return Value**

Type: Boolean  
True when an entity matching the predicate exists, false otherwise.

**Implements**

\texttt{IRepository\texttt{T, TKey Exists(ExpressionFunc\texttt{T, Boolean})}}

### See Also

**Reference**

MongoRepository\texttt{T, TKey} Class  
MongoRepository Namespace
MongoRepository documentation

MongoRepository\(T, TKey\) GetById Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetById(ObjectId)</td>
<td>Returns the T by its given id.</td>
</tr>
<tr>
<td>GetById(TKey)</td>
<td>Returns the T by its given id.</td>
</tr>
</tbody>
</table>

See Also

Reference
MongoRepository\(T, TKey\) Class
MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl
Send comments on this topic to Rob Janssen
**MongoRepository**

* **T**

**GetById** Method (ObjectId)

Returns the T by its given id.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

```csharp
public virtual T GetById(  
    ObjectId id
)
```

### Parameters

**id**

Type: **ObjectId**  
The Id of the entity to retrieve.

### Return Value

Type: **T**  
The Entity T.

### See Also

Reference  
MongoRepository**T**, TKey Class  
GetById Overload  
MongoRepository Namespace
MongoRepository\(T, TKey\)GetById Method (\(TKey\))

Returns the \(T\) by its given id.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

C#  
```csharp
public virtual T GetById(
    TKey id
)
```

### Parameters

- **id**  
  Type: \(TKey\)  
  The Id of the entity to retrieve.

### Return Value

- Type: \(T\)  
  The Entity \(T\).

### Implements

- IRepository\(T, TKey\)GetById(TKey)

### See Also

- Reference  
  - MongoRepository\(T, TKey\) Class  
  - GetById Overload
MongoRepository\tdocumentation

MongoRepository\t

\n\n**MongoRepository\tT, TKey**

 GetEnumerator\tMethod

Returns an enumerator that iterates through a collection.

**Namespace:** MongoRepository
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
<th>C++</th>
<th>F#</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>public virtual</strong> IEnumerator\t&lt;T&gt; GetEnumerator()</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Return Value**

Type: IEnumerator\t<T>

An IEnumerator\t<T> object that can be used to iterate through the collection.

**Implements**

IEnumerable\t<T>GetEnumerator

### See Also

**Reference**

MongoRepository\tT, TKey Class
MongoRepository\tNamespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
MongoRepository\r

documentation

MongoRepository\r

\r

$T$,\r
\r
$TKey$\r

RequestDone Method\r
\r
Lets the server know that this thread is done with a series of related operations.\r
\r
**Namespace:** MongoRepository\r
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)\r
**Version:** 1.6.11.0 (1.6.11.0)\r
\r
### Syntax\r
\r
```c#\r
public virtual void RequestDone()\r
```\r
\r
Implements\r
**IRepository\r
\n$T$, $TKey$RequestDone**\r
\r
### Remarks\r
\r
Instead of calling this method it is better to put the return value of RequestStart in a using statement.\r
\r
### See Also\r
\r
Reference\r
**MongoRepository\r
\n$T$, $TKey$ Class**\r
**MongoRepository Namespace**\r
\r
©2013 - 2015 Rob Janssen / Devcorner.nl\r
\r
Send comments on this topic to Rob Janssen
MongoRepository

**MongoRepository**

*T*, *TKey*

**RequestStart Method**

Lets the server know that this thread is about to begin a series of related operations that must all occur on the same connection. The return value of this method implements IDisposable and can be placed in a using statement (in which case RequestDone will be called automatically when leaving the using statement).

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
<th>C++</th>
<th>F#</th>
<th>Copy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```csharp
public virtual IDisposable RequestStart()
```

**Return Value**

**Type:** IDisposable  
A helper object that implements IDisposable and calls RequestDone() from the Dispose method.

**Implements**

IRepository*T*, TKeyRequestStart

### Remarks

Sometimes a series of operations needs to be performed on the same connection in order to guarantee correct results. This is rarely the case, and most of the time there is no need to call RequestStart/RequestDone. An example of when this might be necessary is when a series of Inserts are called in rapid succession with SafeMode off, and you want to query that data in a consistent manner immediately thereafter (with SafeMode off the writes can
queue up at the server and might not be immediately visible to other
connections). Using RequestStart you can force a query to be on the
same connection as the writes, so the query won't execute until the
server has caught up with the writes.

A thread can temporarily reserve a connection from the connection
pool by using RequestStart and RequestDone. You are free to use
any other databases as well during the request. RequestStart
increments a counter (for this thread) and RequestDone decrements
the counter. The connection that was reserved is not actually
returned to the connection pool until the count reaches zero again.
This means that calls to RequestStart/RequestDone can be nested
and the right thing will happen.

Use the connectionstring to specify the readPreference; add
"readPreference=X" where X is one of the following values: primary,
primaryPreferred, secondary, secondaryPreferred, nearest. See
http://docs.mongodb.org/manual/applications/replication/#read-
preference

See Also

Reference
MongoRepositoryT, TKey Class
MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
MongoRepository $T$, $TKey$ Update Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Update(IEnumerable $T$)</td>
<td>Upserts the entities.</td>
</tr>
<tr>
<td>Update($T$)</td>
<td>Upserts an entity.</td>
</tr>
</tbody>
</table>

See Also

Reference
- MongoRepository $T$, $TKey$ Class
- MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
MongoRepository\textit{T, TKey} Update Method (IEnumerable\textit{T})

Upserts the entities.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
<th>C++</th>
<th>F#</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```csharp
definition public virtual void Update(  
    IEnumerable\textit{T} entities  
)
```

### Parameters

`entities`

- **Type:** System.Collections.Generic\textit{IEnumerable\textit{T}}
- The entities to update.

**Implements**

IRepository\textit{T, TKey} Update(IEnumerable\textit{T})

### See Also

**Reference**

- MongoRepository\textit{T, TKey} Class
- Update Overload
- MongoRepository Namespace
Send comments on this topic to Rob Janssen
MongoRepository\tdocumentation

**MongoRepository**\n\texttt{T, TKeyUpdate Method \,(T)}

Upserts an entity.

**Namespace:** \texttt{MongoRepository}

**Assembly:** \texttt{MongoRepository.Net45 (in MongoRepository.Net45.dll)}

Version: 1.6.11.0 (1.6.11.0)

### Syntax

```csharp
public virtual T Update(
    T entity
)
```

### Parameters

\textit{entity}

Type: \texttt{T}

The entity.

### Return Value

Type: \texttt{T}

The updated entity.

### Implements

\texttt{IRepositoryT, TKeyUpdate(T)}

### See Also

Reference

\texttt{MongoRepositoryT, TKey Class Update Overload}
MongoRepository documentation

MongoRepository $T$, $TKey$ Fields

The MongoRepository $T$, $TKey$ generic type exposes the following members.

## Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>collection</td>
<td>MongoCollection field.</td>
</tr>
</tbody>
</table>

See Also

Reference

MongoRepository $T$, $TKey$ Class
MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
MongoRepository\text{T}, \text{TKey} collection Field

MongoCollection field.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
<th>C++</th>
<th>F#</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>protected internal</code> <code>MongoCollection&lt;T&gt; collection</code></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Field Value  
Type: `MongoCollection<T>`

### See Also

Reference  
MongoRepository\text{T}, \text{TKey} Class  
MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl  
Send comments on this topic to Rob Janssen
MongoRepository documentation

**MongoRepositoryManagerT Class**

Deals with the collections of entities in MongoDB. This class tries to hide as much MongoDB-specific details as possible but it's not 100% *yet*. It is a very thin wrapper around most methods on MongoDB's `MongoCollection` objects.

**Inheritance Hierarchy**

```
System
  Object
  MongoRepository
  MongoRepositoryManagerT, String
    MongoRepositoryMongoRepositoryManagerT
```

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

**Syntax**

```
public class MongoRepositoryManager<T> : MongoRepositoryManager<T, String>
where T : Object, IEntity<string>
```

Type Parameters

T

The type contained in the repository to manage.

The `MongoRepositoryManagerT` type exposes the following members.

**Constructors**
### Name

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MongoRepositoryManagerT</strong></td>
<td>Initializes a new instance of the MongoRepositoryManager class. Uses the Default App/Web.Config connectionstrings to fetch the connectionString and Database name.</td>
</tr>
<tr>
<td><strong>MongoRepositoryManagerT(String)</strong></td>
<td>Initializes a new instance of the MongoRepositoryManager class.</td>
</tr>
<tr>
<td><strong>MongoRepositoryManagerT(String, String)</strong></td>
<td>Initializes a new instance of the MongoRepositoryManager class.</td>
</tr>
</tbody>
</table>

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Exists</strong></td>
<td>Gets a value indicating whether the collection already exists. (Inherited from MongoRepositoryManagerT, TKey.)</td>
</tr>
<tr>
<td><strong>Name</strong></td>
<td>Gets the name of the collection as Mongo uses. (Inherited from MongoRepositoryManagerT, TKey.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Drop</td>
<td>Drops the collection. (Inherited from MongoRepositoryManager TKey.)</td>
</tr>
<tr>
<td>DropAllIndexes</td>
<td>Drops all indexes on the repository. (Inherited from MongoRepositoryManager TKey.)</td>
</tr>
<tr>
<td>DropIndex</td>
<td>Drops specified index on the repository. (Inherited from MongoRepositoryManager TKey.)</td>
</tr>
<tr>
<td>DropIndex</td>
<td>Drops specified indexes on the repository. (Inherited from MongoRepositoryManager TKey.)</td>
</tr>
<tr>
<td>EnsureIndex(String)</td>
<td>Ensures that the desired index exist and creates it if it doesn't exist. (Inherited from MongoRepositoryManager TKey.)</td>
</tr>
<tr>
<td>EnsureIndex(String, Boolean, Boolean, Boolean)</td>
<td>Ensures that the desired index exist and creates it if it doesn't exist.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>EnsureIndexes(IEnumerable&lt;String&gt;)</td>
<td>Ensures that the desired indexes exist and creates them if they don't exist. (Inherited from MongoRepositoryManager&lt;TKey&gt;.)</td>
</tr>
<tr>
<td>EnsureIndexes(IMongoIndexKeys, IMongoIndexOptions)</td>
<td>Ensures that the desired indexes exist and creates them if they don't exist. (Inherited from MongoRepositoryManager&lt;TKey&gt;.)</td>
</tr>
<tr>
<td>EnsureIndexes(IEnumerable&lt;String&gt;, Boolean, Boolean, Boolean)</td>
<td>Ensures that the desired indexes exist and creates them if they don't exist. (Inherited from MongoRepositoryManager&lt;TKey&gt;.)</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash code implementation.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>GetIndexes</td>
<td>Gets the indexes for the repository.</td>
</tr>
<tr>
<td>GetStats</td>
<td>Gets stats for this repository.</td>
</tr>
<tr>
<td>GetTotalDataSize</td>
<td><strong>Obsolete.</strong> Gets the total size for the repository (data + indexes).</td>
</tr>
<tr>
<td>GetTotalStorageSize</td>
<td><strong>Obsolete.</strong> Gets the total storage for the repository (data + indexes).</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <code>Type</code> of the current instance.</td>
</tr>
<tr>
<td>IndexesExists</td>
<td>Tests whether indexes exist.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>IndexExists</td>
<td>Tests whether indexes exist. (Inherited from MongoRepositoryManager TKey.)</td>
</tr>
<tr>
<td>IsCapped</td>
<td>Tests whether the repository is capped. (Inherited from MongoRepositoryManager TKey.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ReIndex</td>
<td>Runs the ReIndex command on this repository. (Inherited from MongoRepositoryManager TKey.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Validate</td>
<td>Validates the integrity of the repository. (Inherited from MongoRepositoryManager TKey.)</td>
</tr>
</tbody>
</table>

Remarks

Entities are assumed to use strings for Id’s.
MongoRepositoryManagerT

Constructor

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MongoRepositoryManagerT</td>
<td>Initializes a new instance of the MongoRepositoryManager class. Uses the Default App/Web.Config connectionstrings to fetch the connectionString and Database name.</td>
</tr>
<tr>
<td>MongoRepositoryManagerT(String)</td>
<td>Initializes a new instance of the MongoRepositoryManager class.</td>
</tr>
<tr>
<td>MongoRepositoryManagerT(String, String)</td>
<td>Initializes a new instance of the MongoRepositoryManager class.</td>
</tr>
</tbody>
</table>

See Also

Reference

MongoRepositoryManagerT Class
MongoRepository Namespace
MongoRepositoryManager<T> Constructor

Initializes a new instance of the MongoRepositoryManager class. Uses the Default App/Web.Config connectionstrings to fetch the connectionString and Database name.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

```csharp
public MongoRepositoryManager()
```

### Remarks

Default constructor defaults to "MongoServerSettings" key for connectionstring.

### See Also

Reference  
- MongoRepositoryManagerT Class  
- MongoRepositoryManagerT Overload  
- MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
MongoRepositoryManagerT Constructor (String)

Initializes a new instance of the MongoRepositoryManager class.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

```csharp
public MongoRepositoryManagerT(string connectionString)
```

### Parameters

- `connectionString`  
  - Type: `System.String`  
  - Connection string to use for connecting to MongoDB.

### See Also

**Reference**  
- MongoRepositoryManagerT Class  
- MongoRepositoryManagerT Overload  
- MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
MongoRepositoryManagerT Constructor (String, String)

Initializes a new instance of the MongoRepositoryManager class.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

```csharp
public MongoRepositoryManager(
    string connectionString,
    string collectionName
);
```

### Parameters

- **connectionString**
  - *Type:* System.String
  - Connectionstring to use for connecting to MongoDB.

- **collectionName**
  - *Type:* System.String
  - The name of the collection to use.

### See Also

- Reference
  - MongoRepositoryManagerT Class
  - MongoRepositoryManagerT Overload
  - MongoRepository Namespace
MongoRepositoryManagerT Properties

The `MongoRepositoryManagerT` generic type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exists</td>
<td>Gets a value indicating whether the collection already exists. (Inherited from <code>MongoRepositoryManagerT, TKey</code>.)</td>
</tr>
<tr>
<td>Name</td>
<td>Gets the name of the collection as <code>Mongo</code> uses. (Inherited from <code>MongoRepositoryManagerT, TKey</code>.)</td>
</tr>
</tbody>
</table>

See Also

Reference

- `MongoRepositoryManagerT Class`
- `MongoRepository Namespace`

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
MongoRepositoryManager\textit{T} Methods

The \textit{MongoRepositoryManager}\textit{T} generic type exposes the following members.

## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drop</td>
<td>Drops the collection. (Inherited from \textit{MongoRepositoryManager}\textit{TKey})</td>
</tr>
<tr>
<td>DropAllIndexes</td>
<td>Drops all indexes on this repository. (Inherited from \textit{MongoRepositoryManager}\textit{TKey})</td>
</tr>
<tr>
<td>DropIndex</td>
<td>Drops specified index on the repository. (Inherited from \textit{MongoRepositoryManager}\textit{TKey})</td>
</tr>
<tr>
<td>DropIndexes</td>
<td>Drops specified indexes on the repository. (Inherited from \textit{MongoRepositoryManager}\textit{TKey})</td>
</tr>
</tbody>
</table>
| EnsureIndex(String)   | Ensures that the desired
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
<th>Inheritance</th>
</tr>
</thead>
<tbody>
<tr>
<td>EnsureIndex(String, Boolean, Boolean, Boolean)</td>
<td>Ensures that the desired index exist and create if it doesn't exist.</td>
<td>(Inherited from MongoRepositoryManager&lt;TKey&gt;.)</td>
</tr>
<tr>
<td>EnsureIndexes(IEnumerable&lt;String&gt;)</td>
<td>Ensures that the desired indexes exist and create if they don't exist.</td>
<td>(Inherited from MongoRepositoryManager&lt;TKey&gt;.)</td>
</tr>
<tr>
<td>EnsureIndexes(IMongoIndexKeys, IMongoIndexOptions)</td>
<td>Ensures that the desired indexes exist and create if they don't exist.</td>
<td>(Inherited from MongoRepositoryManager&lt;TKey&gt;.)</td>
</tr>
<tr>
<td>EnsureIndexes(IEnumerable&lt;String&gt;, Boolean, Boolean, Boolean)</td>
<td>Ensures that the desired indexes exist and create if they don't exist.</td>
<td>(Inherited from MongoRepositoryManager&lt;TKey&gt;.)</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
<td>Inherited From</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try free resources and perform other cleanup operations before it is reclaimed by garbage collection.</td>
<td>Object</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function.</td>
<td>Object</td>
</tr>
<tr>
<td>GetIndexes</td>
<td>Gets the indexes for this repository.</td>
<td>MongoRepositoryManager TKey</td>
</tr>
<tr>
<td>GetStats</td>
<td>Gets stats for this repository.</td>
<td>MongoRepositoryManager TKey</td>
</tr>
<tr>
<td>GetTotalDataSize</td>
<td>Obsolete. Gets the total size for this repository (data + indexes).</td>
<td>MongoRepositoryManager TKey</td>
</tr>
<tr>
<td>GetTotalStorageSize</td>
<td>Obsolete. Gets the total storage for the repository (data + indexes).</td>
<td>MongoRepositoryManager TKey</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>---------------------</td>
<td>--------------------------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object)</td>
<td></td>
</tr>
<tr>
<td>IndexesExists</td>
<td>Tests whether indexes exist. (Inherited from MongoRepositoryManager&lt;TKey&gt;)</td>
<td></td>
</tr>
<tr>
<td>IndexExists</td>
<td>Tests whether indexes exist. (Inherited from MongoRepositoryManager&lt;TKey&gt;)</td>
<td></td>
</tr>
<tr>
<td>IsCapped</td>
<td>Tests whether the repository is capped. (Inherited from MongoRepositoryManager&lt;TKey&gt;)</td>
<td></td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object)</td>
<td></td>
</tr>
<tr>
<td>ReIndex</td>
<td>Runs the ReIndex command on this repository. (Inherited from MongoRepositoryManager&lt;TKey&gt;)</td>
<td></td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object)</td>
<td></td>
</tr>
<tr>
<td>Validate</td>
<td>Validates the integrity of the repository.</td>
<td></td>
</tr>
</tbody>
</table>
(Inherited from MongoRepositoryManager TKeyId.)

See Also

Reference
MongoRepositoryManagerT Class
MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
MongoRepository documentation

MongoRepositoryManager<T, TKey> Class

Deals with the collections of entities in MongoDB. This class tries to hide as much MongoDB-specific details as possible but it's not 100% *yet*. It is a very thin wrapper around most methods on MongoDB's MongoClient objects.

Inheritance Hierarchy

System \rightarrow Object \rightarrow MongoRepository \rightarrow MongoRepositoryManager<T, TKey> \rightarrow MongoRepositoryManager<T, TKey>

Namespace: MongoRepository
Version: 1.6.11.0 (1.6.11.0)

Syntax

```csharp
public class MongoRepositoryManager<T, TKey> : IRepositoryManager
where T : Object, IEntity<TKey>
```

Type Parameters

- **T**
  - The type contained in the repository to manage.

- **TKey**
  - The type used for the entity's Id.

The `MongoRepositoryManager<T, TKey>` type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>MongoRepositoryManagerT</code>, <code>TKey</code></td>
<td>Initializes a new instance of the <code>MongoRepositoryManager</code> class. Uses the Default App/Web.Config connectionstrings to fetch the connectionString and Database name.</td>
</tr>
<tr>
<td><code>MongoRepositoryManagerT</code>, <code>TKey(String)</code></td>
<td>Initializes a new instance of the <code>MongoRepositoryManager</code> class.</td>
</tr>
<tr>
<td><code>MongoRepositoryManagerT</code>, <code>TKey(String, String)</code></td>
<td>Initializes a new instance of the <code>MongoRepositoryManager</code> class.</td>
</tr>
</tbody>
</table>

## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Exists</code></td>
<td>Gets a value indicating whether the collection already exists.</td>
</tr>
<tr>
<td><code>Name</code></td>
<td>Gets the name of the collection as Mongo uses.</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drop</td>
<td>Drops the collection.</td>
</tr>
<tr>
<td>DropAllIndexes</td>
<td>Drops all indexes on this repository.</td>
</tr>
<tr>
<td>DropIndex</td>
<td>Drops specified index on the repository.</td>
</tr>
<tr>
<td>DropIndexes</td>
<td>Drops specified indexes on the repository.</td>
</tr>
<tr>
<td>EnsureIndex(String)</td>
<td>Ensures that the desired index exist and creates it if it doesn't exist.</td>
</tr>
<tr>
<td>EnsureIndex(String, Boolean, Boolean, Boolean)</td>
<td>Ensures that the desired index exist and creates it if it doesn't exist.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>EnsureIndexes(IEnumerable(IEnumerable&lt;\String&gt;))</td>
<td>Ensures that the desired indexes exist and creates them if they don't exist.</td>
</tr>
<tr>
<td>EnsureIndexes(IMongoIndexKeys, IMongoIndexOptions)</td>
<td>Ensures that the desired indexes exist and creates them if they don't exist.</td>
</tr>
<tr>
<td>EnsureIndexes(IEnumerable(IEnumerable&lt;\String&gt;, Boolean, Boolean, Boolean))</td>
<td>Ensures that the desired indexes exist and creates them if they don't exist.</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other</td>
</tr>
</tbody>
</table>
cleanup operations before it is reclaimed by garbage collection. (Inherited from **Object**.)

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td>GetIndexes</td>
<td>Gets the indexes for this repository.</td>
</tr>
<tr>
<td>GetStats</td>
<td>Gets stats for this repository.</td>
</tr>
<tr>
<td>GetTotalDataSize</td>
<td><strong>Obsolete.</strong> Gets the total size for the repository (data + indexes).</td>
</tr>
<tr>
<td>GetTotalStorageSize</td>
<td><strong>Obsolete.</strong> Gets the total storage size for the</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td><code>GetType</code></td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>IndexesExists</code></td>
<td>Tests whether indexes exist.</td>
</tr>
<tr>
<td><code>IndexExists</code></td>
<td>Tests whether indexes exist.</td>
</tr>
<tr>
<td><code>IsCapped</code></td>
<td>Tests whether the repository is capped.</td>
</tr>
<tr>
<td><code>MemberwiseClone</code></td>
<td>Creates a shallow copy of the current <code>Object</code>.  (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>ReIndex</code></td>
<td>Runs the ReIndex command on</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Validate</td>
<td>Validates the integrity of the repository.</td>
</tr>
</tbody>
</table>

**See Also**

Reference

MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
MongoRepositoryManagerT, TKey Constructor

### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MongoRepositoryManagerT, TKey</strong></td>
<td>Initializes a new instance of the MongoRepositoryManager class. Uses the Default App/Web.Config connectionstrings to fetch the connectionString and Database name.</td>
</tr>
<tr>
<td><strong>MongoRepositoryManagerT, TKey(String)</strong></td>
<td>Initializes a new instance of the MongoRepositoryManager class.</td>
</tr>
<tr>
<td><strong>MongoRepositoryManagerT, TKey(String, String)</strong></td>
<td>Initializes a new instance of the MongoRepositoryManager class.</td>
</tr>
</tbody>
</table>

**See Also**

Reference
- MongoRepositoryManagerT, TKey Class
- MongoRepository Namespace
MongoRepositoryManager<T, TKey> Constructor

Initializes a new instance of the MongoRepositoryManager class. Uses the Default App/Web.Config connectionstrings to fetch the connectionString and Database name.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

```csharp
public MongoRepositoryManager()
```

### Remarks

Default constructor defaults to "MongoServerSettings" key for connectionstring.

### See Also

Reference
- MongoRepositoryManager<T, TKey> Class  
- MongoRepositoryManager<T, TKey> Overload  
- MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
MongoRepository documentation

MongoRepositoryManager\(T, TKey\) Constructor (String)

Initializes a new instance of the MongoRepositoryManager class.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
<th>C++</th>
<th>F#</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>publicMongoRepositoryManager(</code></td>
<td><code>public MongoRepositoryManager(</code></td>
<td><code>public MongoRepositoryManager(</code></td>
<td><code>public MongoRepositoryManager(</code></td>
</tr>
<tr>
<td><code>string connectionString</code>)</td>
<td><code>string connectionString</code>)</td>
<td><code>string connectionString</code>)</td>
<td><code>string connectionString</code>)</td>
</tr>
</tbody>
</table>

**Parameters**

*connectionString*

- **Type:** `System.String`  
- Connectionstring to use for connecting to MongoDB.

### See Also

**Reference**

- `MongoRepositoryManager\(T, TKey\)` Class  
- `MongoRepositoryManager\(T, TKey\)` Overload  
- MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
MongoRepositoryManager\(T, TKey\) Constructor (String, String)

Initializes a new instance of the MongoRepositoryManager class.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

```csharp
public MongoRepositoryManager(
    string connectionString,
    string collectionName
)
```

### Parameters

- **connectionString**
  - Type: `System.String`
  - Connectionstring to use for connecting to MongoDB.

- **collectionName**
  - Type: `System.String`
  - The name of the collection to use.

### See Also

- Reference  
  - MongoRepositoryManager\(T, TKey\) Class  
  - MongoRepositoryManager\(T, TKey\) Overload  
  - MongoRepository Namespace
Send comments on this topic to Rob Janssen
MongoRepositoryManager\(T,\ TKey\) Properties

The `MongoRepositoryManager\(T,\ TKey\)` generic type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exists</td>
<td>Gets a value indicating whether the collection already exists.</td>
</tr>
<tr>
<td>Name</td>
<td>Gets the name of the collection as Mongo uses.</td>
</tr>
</tbody>
</table>

### See Also

Reference

- `MongoRepositoryManager\(T,\ TKey\)` Class
- `MongoRepository` Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
MongoRepository documentation

MongoRepositoryManager\(T, TKey\) Exists Property

Gets a value indicating whether the collection already exists.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
<th>C++</th>
<th>F#</th>
</tr>
</thead>
<tbody>
<tr>
<td>public virtual bool Exists { get; }</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Property Value

**Type:** Boolean  
Returns true when the collection already exists, false otherwise.

### Implements

IRepositoryManager\(T, TKey\) Exists

### See Also

**Reference**

MongoRepositoryManager\(T, TKey\) Class  
MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
MongoRepositoryManager\text{T}, TKey

\text{Name\ Property}

Gets the name of the collection as Mongo uses.

\textbf{Namespace:} MongoRepository
Version: 1.6.11.0 (1.6.11.0)

\section*{Syntax}

\begin{center}
\begin{tabular}{llll}
\textbf{C#} & \textbf{VB} & \textbf{C++} & \textbf{F#} \\
\end{tabular}
\end{center}

\begin{center}
\texttt{public virtual string Name \{ get; \}}
\end{center}

\textbf{Property Value}
Type: \texttt{String}
The name of the collection as Mongo uses.

\textbf{Implements}
\texttt{IRepositoryManager\text{T}, TKey\ Name}

\section*{See Also}

\textbf{Reference}
\texttt{MongoRepositoryManager\text{T}, TKey Class}
\texttt{MongoRepository Namespace}

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
The `MongoRepositoryManager<T, TKey>` generic type exposes the following members.

## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drop</td>
<td>Drops the collection.</td>
</tr>
<tr>
<td>DropAllIndexes</td>
<td>Drops all indexes on this repository.</td>
</tr>
<tr>
<td>DropIndex</td>
<td>Drops specified index on the repository.</td>
</tr>
<tr>
<td>DropIndexes</td>
<td>Drops specified indexes on the repository.</td>
</tr>
<tr>
<td>EnsureIndex(String)</td>
<td>Ensures that the desired index exist and creates it if it doesn't</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>EnsureIndex(String, Boolean, Boolean, Boolean)</strong></td>
<td>Ensures that the desired index exist and creates it if it doesn't exist.</td>
</tr>
<tr>
<td><strong>EnsureIndexes(IEnumerable&lt;String&gt;)</strong></td>
<td>Ensures that the desired indexes exist and creates them if they don't exist.</td>
</tr>
<tr>
<td><strong>EnsureIndexes(IMongoIndexKeys, IMongoIndexOptions)</strong></td>
<td>Ensures that the desired indexes exist and creates them if they don't exist.</td>
</tr>
<tr>
<td><strong>EnsureIndexes(IEnumerable&lt;String&gt;, Boolean, Boolean, Boolean)</strong></td>
<td>Ensures that the desired indexes exist and creates them if they don't exist.</td>
</tr>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified object is equal to the current object. (Inherited)</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Finalize               | Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from `Object`.)
<p>| GetHashCode           | Serves as the default hash function. (Inherited from <code>Object</code>.)             |
| GetIndexes             | Gets the indexes for this repository.                                      |
| GetStats               | Gets stats for this repository.                                             |
| GetTotalDataSize       | <strong>Obsolete.</strong> Gets the total size for the                                   |</p>
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>GetTotalStorageSize</code></td>
<td><strong>Obsolete.</strong> Gets the total storage size for the repository (data + indexes).</td>
</tr>
<tr>
<td><code>GetType</code></td>
<td>Gets the Type of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>IndexesExists</code></td>
<td>Tests whether indexes exist.</td>
</tr>
<tr>
<td><code>IndexExists</code></td>
<td>Tests whether indexes exist.</td>
</tr>
<tr>
<td><code>IsCapped</code></td>
<td>Tests whether the repository is capped.</td>
</tr>
<tr>
<td><code>MemberwiseClone</code></td>
<td>Creates a shallow copy of the repository (data + indexes).</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>ReIndex</td>
<td>Runs the ReIndex command on this repository.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Validate</td>
<td>Validates the integrity of the repository.</td>
</tr>
</tbody>
</table>

**See Also**

Reference

MongoRepositoryManagerT, TKey Class

MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
MongoRepositoryManager\text{T}, T\text{KeyDrop} Method

Drops the collection.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

```
public virtual void Drop()
```

**Implements**  
IRepositoryManager\text{T}, T\text{KeyDrop}

### See Also

**Reference**  
MongoRepositoryManager\text{T}, T\text{Key} Class  
MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
**MongoRepositoryManager**

**MongoRepositoryManager**

`TKeyDropAllIndexes Method`

Drops all indexes on this repository.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

**Syntax**

```
public virtual void DropAllIndexes()
```

Implements

`IRepositoryManager`, `TKeyDropAllIndexes`

**See Also**

Reference

`MongoRepositoryManager`, `TKey Class`  
`MongoRepository Namespace`

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
MongoRepository documentation

MongoRepositoryManager<
T,
TKey> DropIndex Method

Drops specified index on the repository.

**Namespace:** MongoRepository
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)
**Version:** 1.6.11.0 (1.6.11.0)

**Syntax**

```csharp
public virtual void DropIndex(
    string keyname
)
```

**Parameters**

`keyname`
Type: `System.String`
The name of the indexed field.

**Implements**
`IRepositoryManager<T, TKey>DropIndex(String)`

**See Also**

**Reference**
MongoRepositoryManager<T, TKey> Class
MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
MongoRepositoryManagerT, TKeyDropIndexes Method

Drops specified indexes on the repository.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
<th>C++</th>
<th>F#</th>
</tr>
</thead>
<tbody>
<tr>
<td>public virtual void DropIndexes(</td>
<td>IEnumerable&lt;string&gt; keynames</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Parameters

**keynames**  
Type: System.Collections.Generic(IEnumerable<string>)  
The names of the indexed fields.

### Implements

IRepositoryManagerT, TKeyDropIndexes(IEnumerable<string>)

### See Also

Reference  
MongoRepositoryManagerT, TKey Class  
MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl  
Send comments on this topic to Rob Janssen
MongoRepositoryManager<T, TKey> EnsureIndex Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>EnsureIndex(String)</code></td>
<td>Ensures that the desired index exist and creates it if it doesn't exist.</td>
</tr>
<tr>
<td><code>EnsureIndex(String, Boolean, Boolean, Boolean)</code></td>
<td>Ensures that the desired index exist and creates it if it doesn't exist.</td>
</tr>
</tbody>
</table>

See Also

Reference
- MongoRepositoryManager<T, TKey> Class
- MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
MongoRepositoryManager{T, TKey} EnsureIndex Method (String)

Ensures that the desired index exist and creates it if it doesn't exist.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

```csharp
public virtual void EnsureIndex(
    string keyname
)
```

### Parameters

**keyname**  
Type: System.String  
The indexed field.

### Remarks

This is a convenience method for EnsureIndexes(IMongoIndexKeys keys, IMongoIndexOptions options). Index will be ascending order, non-unique, non-sparse.
See Also

Reference
MongoRepositoryManagerT, TKey Class
EnsureIndex Overload
MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
MongoRepositoryManager\textit{T}, \textit{TKey}EnsureIndex Method (String, Boolean, Boolean, Boolean)

Ensures that the desired index exist and creates it if it doesn't exist.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

```csharp
public virtual void EnsureIndex(
    string keyname,  
    bool descending,  
    bool unique,  
    bool sparse
)
```

### Parameters

- **keyname**  
  Type: SystemString  
  The indexed field.

- **descending**  
  Type: SystemBoolean  
  Set to true to make index descending, false for ascending.

- **unique**  
  Type: SystemBoolean
Set to true to ensure index enforces unique values.

`sparse`

Type: `System.Boolean`
Set to true to specify the index is sparse.

**Implements**

`IRepositoryManager<T, TKey> EnsureIndex(String, Boolean, Boolean, Boolean)`

**Remarks**

This is a convenience method for `EnsureIndexes(IMongoIndexKeys keys, IMongoIndexOptions options)`.

**See Also**

Reference

`MongoRepositoryManager<T, TKey>`
`EnsureIndex Overload`
`MongoRepository Namespace`

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
### EnsureIndexes Method

#### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>EnsureIndexes(IEnumerable&lt;string&gt;)</code></td>
<td>Ensures that the desired indexes exist and creates them if they don't exist.</td>
</tr>
<tr>
<td><code>EnsureIndexes(IMongoIndexKeys, IMongoIndexOptions)</code></td>
<td>Ensures that the desired indexes exist and creates them if they don't exist.</td>
</tr>
<tr>
<td><code>EnsureIndexes(IEnumerable&lt;string&gt;, Boolean, Boolean, Boolean)</code></td>
<td>Ensures that the desired indexes exist and creates them if they don't exist.</td>
</tr>
</tbody>
</table>

#### See Also
Reference
MongoRepositoryManagerT, TKey Class
MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
MongoRepositoryManager\(T, T\)KeyEnsureIndexes Method (IEnumerable\(\)String)

Ensures that the desired indexes exist and creates them if they don't exist.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

## Syntax

```csharp
public virtual void EnsureIndexes(
    IEnumerable<string> keynames
)
```

### Parameters

**keynames**
  
  Type: `System.Collections.Generic(IEnumerable<string>)`  
  The indexed fields.

### Implements

`IRepositoryManager\(T, T\)KeyEnsureIndexes(IEnumerable\(\)String)`

## Remarks

This is a convenience method for `EnsureIndexes(IMongoIndexKeys keys, IMongoIndexOptions options)`. Index will be ascending order, non-unique, non-sparse.
See Also

Reference
MongoRepositoryManagerT, TKey Class
EnsureIndexes Overload
MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
MongoRepositoryManager<T, TKey> EnsureIndexes Method (IMongoIndexKeys, IMongoIndexOptions)

Ensures that the desired indexes exist and creates them if they don't exist.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

```csharp
public virtual void EnsureIndexes(
    IMongoIndexKeys keys,
    IMongoIndexOptions options
)
```

### Parameters

- **keys**  
  Type: **IMongoIndexKeys**  
  The indexed fields.

- **options**  
  Type: **IMongoIndexOptions**  
  The index options.

### Implements

**IRepositoryManager<T, TKey> EnsureIndexes(IMongoIndexKeys, IMongoIndexOptions)**
Remarks

This method allows ultimate control but does "leak" some MongoDB specific implementation details.

See Also

Reference

MongoRepositoryManagerT, TKey Class
EnsureIndexes Overload
MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl
Send comments on this topic to Rob Janssen
MongoRepositoryManager<T, TKey> EnsureIndexes Method (IEnumerable<String>, Boolean, Boolean, Boolean)

Ensures that the desired indexes exist and creates them if they don't exist.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

```csharp
public virtual void EnsureIndexes(
    IEnumerable<string> keynames,
    bool descending,
    bool unique,
    bool sparse
)
```

### Parameters

- **keynames**  
  Type: `System.Collections.Generic(IEnumerable<String>)`  
  The indexed fields.

- **descending**  
  Type: `System.Boolean`  
  Set to true to make index descending, false for ascending.

- **unique**
Type: `System.Boolean`
Set to true to ensure index enforces unique values.

*sparse*
Type: `System.Boolean`
Set to true to specify the index is sparse.

**Implements**
`IRepositoryManager<T, TKey>`, `EnsureIndexes(IEnumerable<String>, Boolean, Boolean, Boolean)`

**Remarks**
This is a convenience method for `EnsureIndexes(IMongoIndexKeys keys, IMongoIndexOptions options)`.

**See Also**
Reference
`MongoRepositoryManager<T, TKey>`, `TKey Class
EnsureIndexes Overload
MongoRepository Namespace`
MongoRepositoryManager<T, TKey> GetIndexes Method

Gets the indexes for this repository.

**Namespace:**  MongoRepository
**Assembly:**  MongoRepository.Net45 (in MongoRepository.Net45.dll)
**Version:**  1.6.11.0 (1.6.11.0)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
<th>C++</th>
<th>F#</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>public virtual</strong> GetIndexesResult GetIndexes()</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Return Value**

**Type:**  GetIndexesResult

Returns the indexes for this repository.

**Implements**

IRepositoryManager<T, TKeyGetIndexes

### See Also

**Reference**

MongoRepositoryManager<T, TKey Class
MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
MongoRepositoryManager{T, TKey} GetStats Method

Gets stats for this repository.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
<th>C++</th>
<th>F#</th>
</tr>
</thead>
<tbody>
<tr>
<td>public virtual CollectionStatsResult GetStats()</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Return Value**

Type: **CollectionStatsResult**  
Returns a CollectionStatsResult.

**Implements**

IRepositoryManager{T, TKey}

### Remarks

You will need to reference MongoDB.Driver.

### See Also

**Reference**

MongoRepositoryManager{T, TKey} Class  
MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
MongoRepositoryManager<T, TKey> GetTotalDataSize Method

Note: This API is now obsolete.

Gets the total size for the repository (data + indexes).

Namespace: MongoRepository
Version: 1.6.11.0 (1.6.11.0)

Syntax

```
[ObsoleteAttribute("This method will be removed in the next version of the driver")]
public virtual long GetTotalDataSize()
```

Return Value
Type: Int64
Returns total size for the repository (data + indexes).

Implements
IRepositoryManager<T, TKey> GetTotalDataSize

See Also

Reference
MongoRepositoryManager<T, TKey> Class
MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl
Send comments on this topic to Rob Janssen
MongoRepositoryManager<T, TKey> GetTotalStorageSize Method

Note: This API is now obsolete.

Gets the total storage size for the repository (data + indexes).

Namespace: MongoRepository
Version: 1.6.11.0 (1.6.11.0)

Syntax

C#   VB   C++   F#   Copy

```csharp
[ObsoleteAttribute("This method will be removed in the next version of the driver")]
public virtual long GetTotalStorageSize()
```

Return Value

Type: Int64

Returns total storage size for the repository (data + indexes).

Implements

IRepositoryManager<T, TKey>GetTotalStorageSize

See Also

Reference

MongoRepositoryManager<T, TKey> Class
MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl
Send comments on this topic to Rob Janssen
MongoRepositoryManager\text{\textbackslash}T, TKeyIndexesExists Method

Tests whether indexes exist.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
<th>C++</th>
<th>F#</th>
</tr>
</thead>
</table>
| public virtual bool IndexesExists( 
  IEnumerable\text{\textbackslash}string\text{\textbackslash>} keynames 
) |

### Parameters

*keynames*

Type: System\.Collections\.Generic\.IEnumerable\.String  
The indexed fields.

### Return Value

Type: Boolean  
Returns true when the indexes exist, false otherwise.

### Implements

IRepositoryManager\text{\textbackslash}T, TKeyIndexesExists(IEnumerable\text{\textbackslash}String)

### See Also

Reference

MongoRepositoryManager\text{\textbackslash}T, TKey Class  
MongoRepository Namespace
MongoRepository documentation

MongoRepositoryManager{T, TKey}IndexExists Method

Tests whether indexes exist.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
<th>C++</th>
<th>F#</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```csharp
public virtual bool IndexExists(
    string keyname
)
```

### Parameters

**keyname**  
Type: System.String  
The indexed fields.

### Return Value

Type: Boolean  
Returns true when the indexes exist, false otherwise.

**Implements**  
IRepositoryManager{T, TKey}IndexExists(String)

### See Also

**Reference**  
MongoRepositoryManager{T, TKey} Class  
MongoRepository Namespace
MongoRepositoryManager{T, TKey} IsCapped Method

Tests whether the repository is capped.

**Namespace:** MongoDBRepository  
**Assembly:** MongoDBRepository.Net45 (in MongoDBRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
<th>C++</th>
<th>F#</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>public virtual bool IsCapped()</code></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Return Value

**Type:** Boolean  
Returns true when the repository is capped, false otherwise.

### Implements

`IRepositoryManager{T, TKey}`

### See Also

Reference

- `MongoRepositoryManager{T, TKey} Class`  
- `MongoRepository Namespace`

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to [Rob Janssen](mailto:info@devcorner.nl)
MongoRepository documentation

MongoRepositoryManager<

TKeyReIndex Method

Runs the ReIndex command on this repository.

**Namespace:** MongoRepository

**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)

**Version:** 1.6.11.0 (1.6.11.0)

### Syntax

```
public virtual void ReIndex()
```

Implements

IRepositoryManager<T, TKeyReIndex

### See Also

Reference

MongoRepositoryManager<T, TKey Class

MongoRepository Namespace

©2013 - 2015 Rob Janssen / Devcorner.nl

Send comments on this topic to Rob Janssen
MongoRepositoryManager<T, TKey> Validate Method

Validates the integrity of the repository.

**Namespace:** MongoRepository  
**Assembly:** MongoRepository.Net45 (in MongoRepository.Net45.dll)  
**Version:** 1.6.11.0 (1.6.11.0)

**Syntax**

```csharp
public virtual ValidateCollectionResult Validate()
```

**Return Value**

- **Type:** ValidateCollectionResult  
  - Returns a ValidateCollectionResult.

**Remarks**

- You will need to referenceMongoDb.Driver.

**See Also**

- Reference  
  - MongoRepositoryManager<T, TKey> Class  
  - MongoRepository Namespace
Send comments on this topic to Rob Janssen