|  |
| --- |
|  |
| Service Actions Document |
|  |
|  |
| ©MetaShare Inc. |
| **8/22/2022** |

|  |
| --- |
| How To Use Service |

|  |  |  |
| --- | --- | --- |
| **Date** | **Modifier** | **Description** |
| 2022-08-22 | MDDGroup | first draft |
|  |  |  |
|  |  |  |

目录

[1.1 Instruction 1](#_Toc112078020)

[1.2 Specific Usage 1](#_Toc112078021)

[1.2.1 Entity definition 1](#_Toc112078022)

[1.2.2 Service definition 1](#_Toc112078023)

[1.2.3 Service’s Methods and properties included by default 2](#_Toc112078024)

[1.2.4 ServiceInterface definition 2](#_Toc112078025)

[1.2.5 ServiceInterface’s Methods included by default 3](#_Toc112078026)

[1.2.6 Use of partial Dao methods in Service 5](#_Toc112078027)

[1.2.7 Use of Dao in Service 5](#_Toc112078028)

[1.2.8 Use of transactions in Service 6](#_Toc112078029)

[1.2.9 RegisterService 6](#_Toc112078030)

[1.2.10 Use of Service in other items 7](#_Toc112078031)

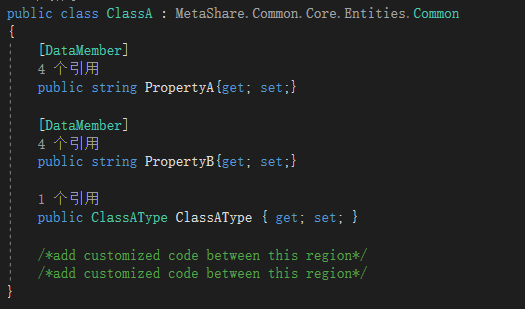
## Instruction

MMC generates project's Service description and usage instructions;

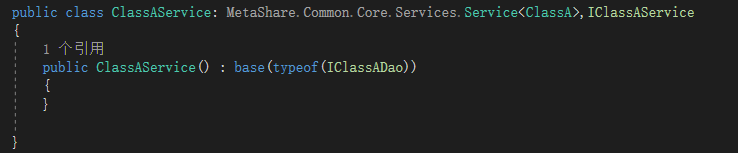
The base is on CommonCore's Service description and usage instructions.

## Specific Usage

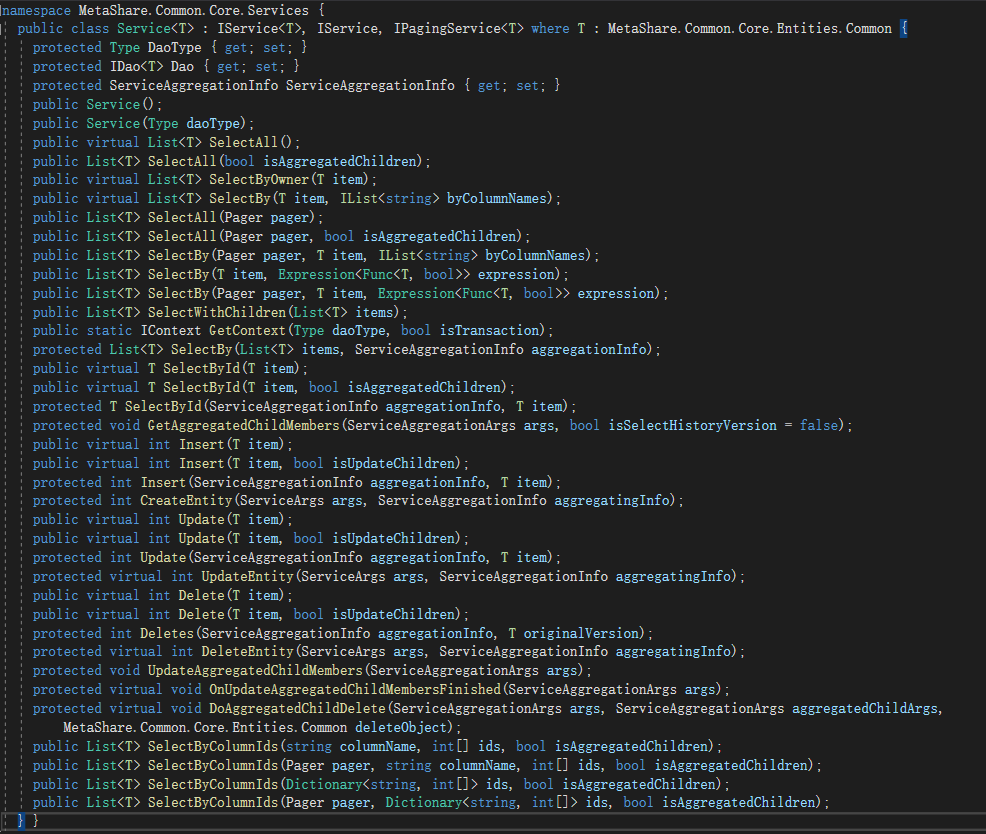
### Entity definition



### Service definition

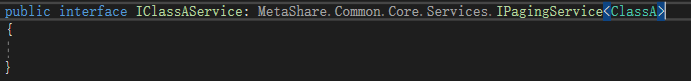


### Service’s Methods and properties included by default

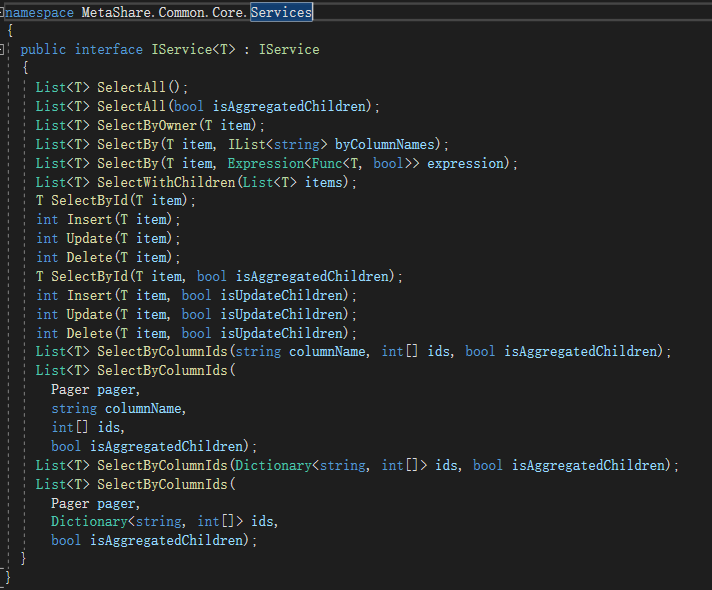
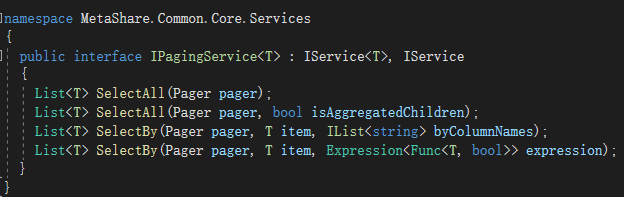


New methods can be rewritten or defined on this basis to meet usage requirements.

### ServiceInterface definition



### ServiceInterface’s Methods included by default



List<T> SelectAll(Pager pager);

--- Query the data collection by all data page

List<T> SelectAll(Pager pager, bool isAggregatedChildren);

--- Query the data set by paging all data. You can configure whether to query sub-attribute data

List<T> SelectBy(Pager pager, T item, IList<string> byColumnNames);

--- Query the data collection through column condition paging

List<T> SelectBy(Pager pager, T item, Expression<Func<T, bool>> expression);

--- Query the data collection by paging through Lambda expression conditions

List<T> SelectAll();---Querying all Data

List<T> SelectAll(bool isAggregatedChildren);

--- Query all data. You can set whether to query sub-attribute data

List<T> SelectByOwner(T item);

--- Query the data through Owner, the method is not implemented yet

List<T> SelectBy(T item, IList<string> byColumnNames);--- Query data by column criteria

List<T> SelectBy(T item, Expression<Func<T, bool>> expression);

--- Query data through Lambda expression conditions

List<T> SelectWithChildren(List<T> items);--- Query sub-attribute data

T SelectById(T item);--- Query data by Id

int Insert(T item);--- Insert data

int Update(T item);--- update data

int Delete(T item);--- deleted data

T SelectById(T item, bool isAggregatedChildren);

--- Query data by Id. You can configure whether to set query sub-attribute data

int Insert(T item, bool isUpdateChildren);

--- Insert data. You can configure whether to insert child attribute data at the same time

int Update(T item, bool isUpdateChildren);

--- Update data. You can configure whether to update sub-attribute data at the same time

int Delete(T item, bool isUpdateChildren);

--- Delete data. You can configure whether to delete child attribute data at the same time

List<T> SelectByColumnIds(string columnName, int[] ids, bool isAggregatedChildren);

--- By querying data in the data set of columns, you can configure whether to set to query sub-attribute data

List<T> SelectByColumnIds(Pager pager, string columnName, int[] ids, bool isAggregatedChildren);

--- To query paging data through the data set of columns, you can configure whether to set to query sub-attribute data

List<T> SelectByColumnIds(Dictionary<string, int[]> ids, bool isAggregatedChildren);

--- By querying data in the column data set dictionary, you can configure whether to set to query sub-attribute data

List<T> SelectByColumnIds(Pager pager, Dictionary<string, int[]> ids, bool isAggregatedChildren);

--- Through the column data set dictionary to query paging data, you can configure whether to set to query sub-attribute data

### Use of partial Dao methods in Service

#### List<T> SelectBy(IContext context, T item, IList<string> byColumnNames);

1. List<ClassA> classAs = this.Dao.SelectBy(context, new ClassA { JOB\_UNIQUE\_ID = jobUniqueId }, new List<string> { "JOB\_UNIQUE\_ID" });
2. List<ClassA> items = this.Dao.SelectBy(context, new ClassA { ClassAType = new ClassAType{ Id = classATypeId } },new List<string> { "ClassATypeId" });

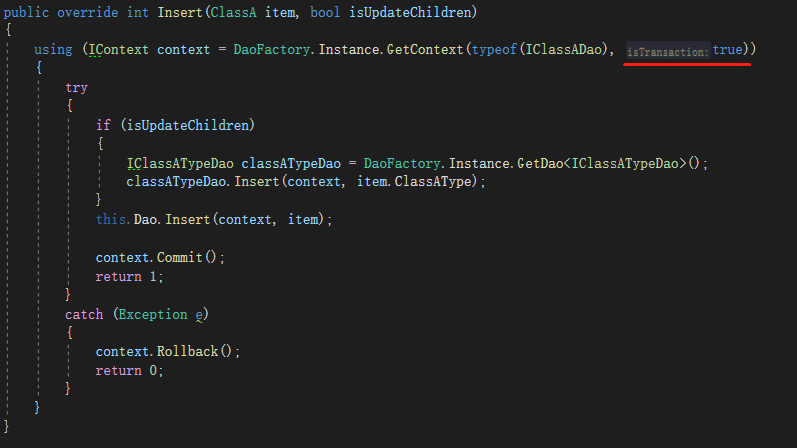
#### List<T> SelectBy(IContext context, Expression<Func<T, bool>> expression);

1. List<ClassA> classAs1 = this.Dao.SelectBy(context, classA => classA.Name == "classA1" );
2. List<ClassA> classAs2 = this.Dao.SelectBy(context, classA => classA.Name.Contains("classA1") );
3. List<ClassA> classAs3 = this.Dao.SelectBy(context, classA => classA.ClassAType.Id == 1);

### Use of Dao in Service

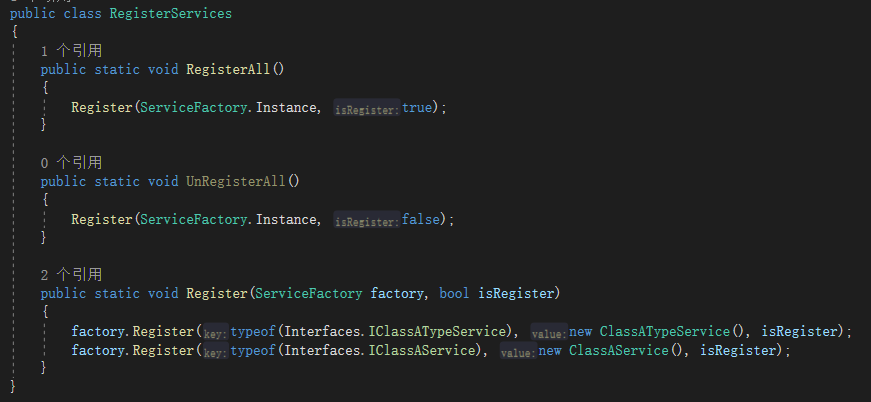
as following picture shows

### Use of transactions in Service



Note: Avoid deadlocks in use.

### RegisterService



### Use of Service in other items

