**ORM Dapper**

**前言:**

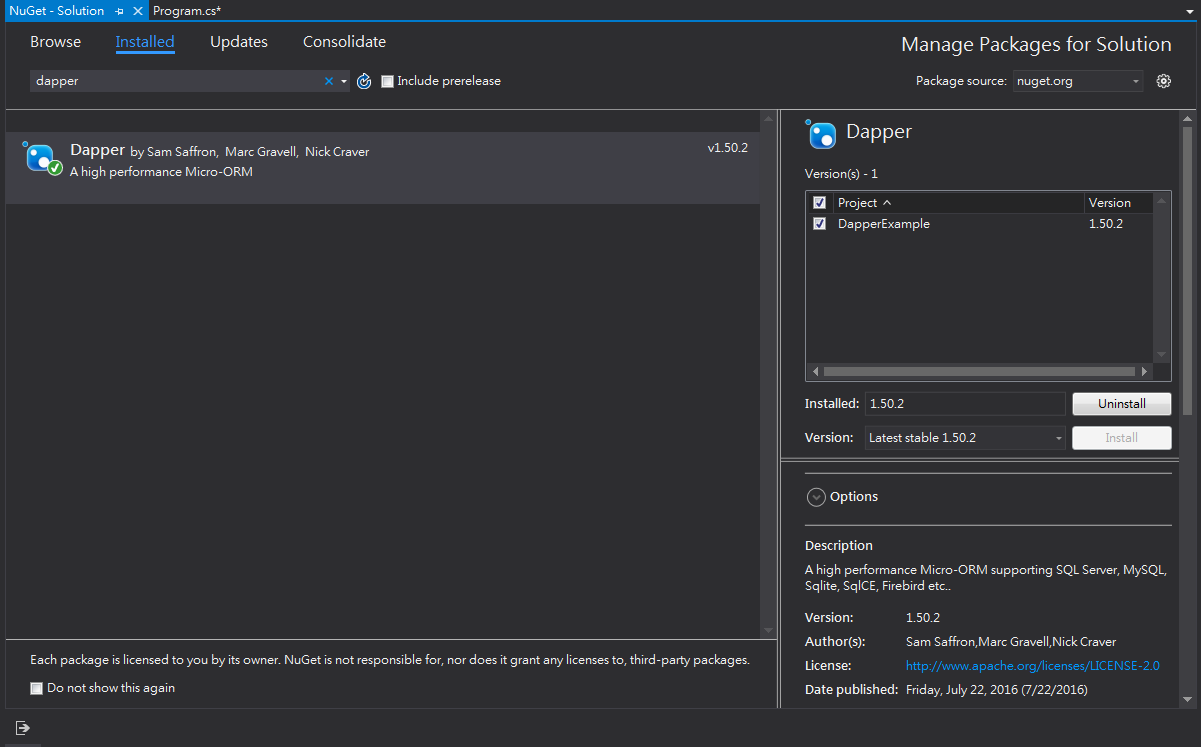
輕量化的Dapper ORM，透過SQL語法，可以快速地從資料庫存資料。

**Dapper特點:**

1. 檔案才 65～75 KB 左右（v1.13），名副其實的微型 ORM  
2. 效能好  
3. 不需要定義POCO物件，直接SELECT結果轉成.NET物件集合  
4. 可宣告及傳入具名參數

**實際操作:**

1. NuGet安裝套件



2. 建立對應資料庫的Model

**public** **class** customers

{

**public** **string** CustomerID { **get**; **set**; }

**public** **string** ContactName { **get**; **set**; }

**public** **string** CompanyName { **get**; **set**; }

**public** **string** City { **get**; **set**; }

**public** **string** Phone { **get**; **set**; }

**public** IEnumerable<Orders> Orders { **get**; **set**; }

}

**public** **class** Orders

{

**public** **string** OrderID { **get**; **set**; }

**public** **string** CustomerID { **get**; **set**; }

**public** **string** OrderDate { **get**; **set**; }

**public** **string** Freight { **get**; **set**; }

}

3. 建立DB連線

**var** ConString = "Data Source=127.0.0.1;" + "Initial Catalog=Northwind;" +"User id=sa;" +"Password=g61109;";

**using** (**var** cn = **new** SqlConnection(ConString))

{

cn.Open();

}

4. 基本查詢和參數查詢

**var** ConString = "Data Source=127.0.0.1;" + "Initial Catalog=Northwind;" +"User id=sa;" +"Password=g61109;";

**using** (**var** cn = **new** SqlConnection(ConString))

{

cn.Open();

//範例一：查詢資料

**string** Query01 = "select \* from Customers";

**var** customers = cn.Query<customers>(Query01);

//範例二：參數式查詢

**string** Query02 = "select \* from Customers where CustomerID = @c\_ID";

**var** para = **new** { c\_ID = "ALFKI" };

**var** customer02 = cn.Query<customers>(Query02, para);

}

5. 執行非查詢類的 SQL 命令，可使用 Execute 擴充方法

//執行非查詢類的 SQL 命令，可使用 Execute 擴充方法

**var** Insert = "insert into Customers (CustomerID, ContactName, CompanyName, City, Phone)" +

" values (@CustomerID, @ContactName, @CompanyName, @City, @Phone)";

**var** newCust = **new** customers()

{

CustomerID = "Z001",

ContactName = "Mike",

CompanyName = "MikeSoft",

City = "台北市",

Phone = "12345678"

};

**int** rowsChanged = cn.Execute(Insert, newCust);

**一對多(one to many)查詢:**

1. 自行新增母子Mapping方法

**public** **class** EnittyOneToManyMapper<TP, TC, TPk>

{

**private** **readonly** IDictionary<TPk, TP> \_lookup = **new** Dictionary<TPk, TP>();

**public** Action<TP, TC> AddChildAction { **get**; **set**; }

**public** Func<TP, TPk> ParentKey { **get**; **set**; }

**public** **virtual** TP Map(TP parent, TC child)

{

TP entity;

**var** found = **true**;

**var** primaryKey = ParentKey(parent);

**if** (!\_lookup.TryGetValue(primaryKey, **out** entity))

{

\_lookup.Add(primaryKey, parent);

entity = parent;

found = **false**;

}

AddChildAction(entity, child);

**return** !found ? entity : **default**(TP);

}

}

1. **搜尋母子資料後執行方法EnittyOneToManyMapper**

//範例三:一對多查詢

**string** Query03 = "SELECT \* FROM Customers c JOIN Orders o ON c.CustomerId = o.CustomerId ORDER BY c.CustomerId";

**var** mapper = **new** EnittyOneToManyMapper<customers, Orders, **string**>()

{

AddChildAction = (c, o) =>

{

**if** (c.Orders == **null**)

c.Orders = **new** List<Orders>();

c.Orders = c.Orders.Concat(**new**[] { o });

},

ParentKey = (c) => c.CustomerID

};

**var** customer03 = cn.Query<customers, Orders, customers>(Query03, mapper.Map, splitOn: "OrderId")

.Where(y => y != **null**);

**參考資料:**

<http://huan-lin.blogspot.com/2014/03/a-micro-orm-dapper.html>  
[**http://blog.darkthread.net/post-2014-05-15-dapper.aspx**](http://blog.darkthread.net/post-2014-05-15-dapper.aspx)