

2. ORM 操作

1. 目前提供支援的資料庫有Sql Server、DB2
2. 繼承BaseController、BaseService，就會有共用一般資料庫連線Db可使用
3. 繼承DataBase

```
5 個參考 | ben, 9 天前 | 1 位作者, 1 項變更
public class SAVDataBase : DataBase
{
    4 個參考 | ben, 9 天前 | 1 位作者, 1 項變更
    public SAVDataBase(string connection)
        : base(connection, AppConfig.key)
    {
    }
}

2 個參考 | A9839, 1 天前 | 1 位作者, 1 項變更
public class DB2DataBase : DataBase
{
    1 個參考 | A9839, 1 天前 | 1 位作者, 1 項變更
    public DB2DataBase(string connection)
        : base(connection, AppConfig.key)
    {
    }
}
```

4. 建立DataBase及操作DB2

```
try
{
    string sql = @"SELECT (A.PTRNO_PROGRAM || ' ' || A.PTRNO_RESOURCE) AS TRANCODE,
        COALESCE(D.TRANO_SUBM_PGMN,(A.PTRNO_PROGRAM || ' ' || A.PTRNO_RESOURCE)) AS TRANCODEDESC
    FROM DBFUNDNR.FFDPTRNO A
    LEFT JOIN DBFUNDNR.FSCVVERTO P ON A.PTRNO_PROGRAM=P.VERTO_PROGRAM AND A.PTRNO_RESOURCE=P.VERTO_RESOURCE
    LEFT JOIN DBFUNDNR.FFDTRANO D ON A.PTRNO_PROGRAM=D.TRANO_PROGRAM AND A.PTRNO_RESOURCE=D.TRANO_RESOURCE
    WHERE A.PTRNO_POLICY_NO='K121406404' AND A.PTRNO_POLICY_SEQ=01 AND A.PTRNO_ID_DUP=''
    AND P.VERTO_FD_YN IN ('0','1','2','3','4','5') AND NOT (A.PTRNO_PROGRAM='BFD6300' AND A.PTRNO_RESOURCE='')";
    var db = new DB2DataBase("DB2");
    data = db.GetListPaged<DemoModel1>(2, 20, sql, null, "A.PTRNO_PROGRAM || ' ' || A.PTRNO_RESOURCE");
}
catch (Exception ex)
{
    ViewBag.Output = ex.Message;
}
```

5. Model範例如下

-

```

[Table("Product")]
[Order("Seq Desc")]
17 個參考 | A9839, 1 天前 | 2 位作者, 4 項變更
public class ProductModel
{
    [Key]
    [Fubon.Orm.ReadOnly(true)]
    [I18NDisplayName("Product", "Seq")]
    0 個參考 | A9839, 1 天前 | 2 位作者, 2 項變更
    public Int64? Seq { get; set; }

    [Key]
    [Column("ID")]
    [I18NDisplayName("Product", "ID")]
    3 個參考 | A9839, 6 天前 | 2 位作者, 2 項變更
    public Guid ID { get; set; }

    [Required]
    [I18NDisplayName("Product", "Name")]
    [I18NMaxLength(50, "MaxLength")]
    0 個參考 | ben, 9 天前 | 1 位作者, 1 項變更
    public string Name { get; set; }

    [Required]
    [I18NDisplayName("Product", "Price")]
    0 個參考 | ben, 9 天前 | 1 位作者, 1 項變更
    public decimal? Price { get; set; }

    [I18NDisplayName("Product", "CreateDate")]
    6 個參考 | ben, 9 天前 | 1 位作者, 1 項變更
    public DateTime CreateDate { get; set; }

    [I18NDisplayName("Product", "CreateUser")]
    [I18NMaxLength(50, "MaxLength")]
    6 個參考 | ben, 9 天前 | 1 位作者, 1 項變更
    public string CreateUser { get; set; }

    0 個參考 | ben, 9 天前 | 1 位作者, 1 項變更
    public byte[] UpFile { get; set; }
}

```

- Table：需與資料表名稱相同
- Order：預設排序，EX：[Order("Name Desc, Price Asc")]
- Key：可以在多個Column上面定義

- Column：表格欄位名稱
- Fubon.Orm.ReadOnly：唯獨，在insert或update，會忽略此欄位的更新
- I18NDisplayName：配合多國語系，第一個參數請輸入Table Name，第二個參數為Column Name，Resource命名代碼為 Table_Column
- I18NMaxLength：配合多國語系，最大長度限制
- I18NLength：配合多國語系，最大、最小長度限制

6. Sql Command

```
//取得單一
varmodel=Db.Get<UserInfoModel>(id);

//取得全部
varlist=Db.GetList<UserInfoModel>();

//新增UserInfo
varuserinfo2=newUserInfoModel();
userinfo2.Name=" test2";
userinfo2.UserID=" test2";
userinfo2.Identity=0;
userinfo2.Gender=0;
userinfo2.Education=0;
userinfo2.Tel_M="096XXX56";
userinfo2.Email="teXxt2@gmail.com";
userinfo2.Pwd="12345678";
Db.Insert(userinfo2);

//修改UserInfo
vartUser2=Db.GetList<UserInfoModel>(userinfo2.Email).First(); tUser2.Name="tesXXdate";
tUser2.UserID=" test2Upd";
tUser2.Identity=0;
tUser2.Gender=0;
tUser2.Education=0;
Db.Update(tUser2);

//刪除
tUser2=Db.GetList<UserInfoModel>(userinfo2.Email).First();
Db.Delete(tUser2);
```

7. Transaction

```
Db.BeginTransaction();

try {
varuser=newUserInfoModel();
user.UserKey=Guid.NewGuid();
user.Name=" test1234";
user.UserID=" A1XXX56789";
user.Tel_M="09XXX3456";
user.Email="tranXXXXn@gmail.com";
```

```

user.Pwd="12345678";

Db.Insert(user);

user.Name="testupdate";

Db.Update(user);

Db.CompleteTransaction();

}

catch

{

Db.AbortTransaction();

}

finally{}

```

8. Get與GetOne之差異：

- Get：當Model只有一個Key，且Key為整數或GUID才可使用
- GetOne：當Model有一個Key以上都可以使用

9. Update時，需注意Model的所有Key，是否都有值，否則會有無法更新的情況。
10. 當要使用join，可以自己組Sql QueryString或使用Linq，依據個人習慣及狀況而定。
11. GetListPaged

```

#region ORM + Where
string conditions = "where l=1 ";
string proName = GetFormValue("proName");
if (proName.IsNotNullAndTrimEmpty()) conditions += string.Format(" and Name like '{0}%' ", proName.Replace("'", "''"));
string proPrice = GetFormValue("proPrice");
if (proPrice.IsNotNullAndTrimEmpty()) conditions += string.Format(" and Price={0} ", proPrice.Replace("'", "''"));
var list = ProductDao.GetListPaged(Pagination.Page, Pagination.PerPage, conditions, Pagination.OrderBy);
#endregion

#region ORM + Object
string proName = GetFormValue("proName");
string proPrice = GetFormValue("proPrice");
var list = ProductDao.GetListPaged(Pagination.Page, Pagination.PerPage, new { Name = proName, Price = proPrice }, Pagination.OrderBy);
#endregion

#region SqlCommand + Object
string proName = GetFormValue("proName");
string proPrice = GetFormValue("proPrice");
string sql = @"Select ID,Seq,Name,Price,CreateDate,CreateUser from Product
Where Name like @Name ";
if (proPrice.IsNotNullAndTrimEmpty())
    sql += "And Price=@Price ";
var list = ProductDao.GetListPaged(Pagination.Page, Pagination.PerPage, sql, new { Name = "%" + proName + "%", Price = proPrice },
, Pagination.OrderBy);
#endregion

```