|  |
| --- |
| ClassLibrary建立 |
| |  | | --- | | **建立檔案** | | 檔案 =>專案 =>新增 =>類別庫(.NET Framework) | |

|  |
| --- |
| ClassLibrary宣告使用 |
| |  | | --- | | **建立檔案** | | 1. ClassLibrary專案先重建方案(可建立D: \OD\_ClassLibrary\OD\_ClassLibrary\bin\Debug\OD\_ClassLibrary.dll)  2.需要使用OD\_ClassLibrary的專案中  參考 =>加入參考 => 瀏覽 =>D: \OD\_ClassLibrary\OD\_ClassLibrary\bin\Debug\OD\_ClassLibrary.dll  3.在程式中宣告  using OD\_ClassLibrary;  public Class1 class1 = new Class1(); // Class1 是OD\_ClassLibrary中 namespace的名稱 | |

|  |
| --- |
| ClassLibrary專案內碼更新 |
| |  | | --- | | **建立檔案** | | 1. ClassLibrary專案先重建方案(可建立D: \OD\_ClassLibrary\OD\_ClassLibrary\bin\Debug\OD\_ClassLibrary.dll)  2.需要使用OD\_ClassLibrary的專案中  參考 =>加入參考 => 瀏覽 =>D: \OD\_ClassLibrary\OD\_ClassLibrary\bin\Debug\OD\_ClassLibrary.dll | |

|  |
| --- |
| OD\_ClassLibrary |
| |  | | --- | | **List頁數**   1. **宣告使用Numberofpages<T>(List<T> data,int number,int now\_page)**   **(1). List<T> data 需要設定的資料清單**  **(2). Number 每頁顯示資料筆數**  **(3). Now\_page 目前顯示頁數**   1. **T型別是Numberofpages<T>設定** 2. **data.Skip(0).Take(10) 顯示1-10的資料** | | public List<T> Numberofpages <T>(List<T> data,int number,int now\_page) where T :new ()  {  List<T> return\_list = new List<T>();  return\_list = data.Skip(number\*(now\_page-1)).Take(number+number\*(now\_page-1)).ToList();  if (return\_list.Count < number)  {  for(int a = return\_list.Count; a < number; a++)  {  return\_list.Add(new T());  }  }  return return\_list;  } | | **DateTable 轉成List** | | public List<T> DataTableToListWithJson<T>(DataTable dt)  {  string json = JsonConvert.SerializeObject(dt); // 將 DataTable 轉為 JSON  return JsonConvert.DeserializeObject<List<T>>(json); // 將 JSON 轉為 T 的列表  } | | **SQL資料查詢FUNCTION** | | public DataTable SQL\_select\_exce(string sqlstring)  {  string connectionString = "連資料庫路徑";  SqlDataReader reader;  DataTable data = new DataTable();  using (SqlConnection conn = new SqlConnection(connectionString))  {  conn.Open();  using (SqlCommand cmd = new SqlCommand(sqlstring, conn))  {  reader = cmd.ExecuteReader();//讀取回傳的資料  data.Load(reader);  }  }  return data;  } | |

|  |
| --- |
| SQL資料查詢 |
| |  | | --- | | **Model 呼叫SQL資料查詢語法** | | public class Production  {  public string type\_id { get; set; } //編號名稱  public string type\_name { get; set; } //編號名稱  public string product\_id { get; set; } //編號  public string name { get; set; }//商品姓名  public int number { get; set; }//庫存數量  public int price { get; set; }//售貨價格  public DateTime? dr\_date { get; set; }  public DateTime? dt\_date { get; set; }  }  public class product\_homepage\_model  {  public string type\_id { get; set; }//類型編號  public string product\_name { get; set; } //產品名稱  public page page { get; set; }  }  public class page  {  public int numbers { get; set; }//顯示數量  public int now\_page { get; set; }//目前頁數  public int end\_page { get; set; }//最後一頁數  }  public List<Production> Products\_list(string types\_id, string sreach)  {  List<Production> return\_list = new List<Production>();  string sql\_list = string.Empty;  //如果 true 參數為 value 或空字串 ("")，則為 null，否則為 false。  if (!String.IsNullOrEmpty(types\_id))//有資料  {  sql\_list += $"and p.type\_id='{types\_id}'";  }  if (!String.IsNullOrEmpty(sreach))//有資料  {  sql\_list += $"and p.name='{sreach}'";  }  sql\_list = $@"select p.product\_id,p.name,p.number,p.price from orders.dbo.Production as p with (nolock)  where 1=1 and p.number>1 " + sql\_list;  //資料讀取  return\_list = class1.DataTableToListWithJson<Production>(class1.SQL\_select\_exce(sql\_list));  return return\_list;  } | | **Controller** | | public ActionResult Product\_homepage\_Index()  {  OD\_ViewModels ViewModels = new OD\_ViewModels();  OD\_Models Models = new OD\_Models();  ViewModels.product\_homepage\_model = new product\_homepage\_model();  ViewModels.product\_homepage\_model.page = new page();  ViewModels.Types\_list = Models.Types\_list();  ViewModels.Products\_list = Models.Products\_list(null, null);  #region 頁數  ViewModels.product\_homepage\_model.page.now\_page = 1;  ViewModels.product\_homepage\_model.page.numbers = 7;  //值相除後如若有小數點將會無條件進位  var a = Math.Ceiling(decimal.Parse(ViewModels.Products\_list.Count.ToString()) / decimal.Parse(ViewModels.product\_homepage\_model.page.numbers.ToString()));  ViewModels.product\_homepage\_model.page.end\_page = (int)a;  #endregion  ViewModels.Products\_list = class1.Numberofpages<Production>(ViewModels.Products\_list, ViewModels.product\_homepage\_model.page.numbers, ViewModels.product\_homepage\_model.page.now\_page);  return View(ViewModels);  }  [HttpPost]  public ActionResult Product\_homepage\_Index(OD\_ViewModels ajs)  {  OD\_ViewModels ViewModels = new OD\_ViewModels();  OD\_Models Models = new OD\_Models();  ViewModels.product\_homepage\_model = new product\_homepage\_model();  ViewModels.product\_homepage\_model.page = ajs.product\_homepage\_model.page;  ViewModels.Types\_list = Models.Types\_list();  ViewModels.Products\_list = Models.Products\_list(ajs.product\_homepage\_model.type\_id, ajs.product\_homepage\_model.product\_name);  #region 頁數  //值相除後如若有小數點將會無條件進位  var a = Math.Ceiling(decimal.Parse(ViewModels.Products\_list.Count.ToString()) / decimal.Parse(ViewModels.product\_homepage\_model.page.numbers.ToString()));  /如若最後一頁數與原值不相等需要重新設定預設值  if(ViewModels.product\_homepage\_model.page.end\_page != (int)a)  {  ViewModels.product\_homepage\_model.page.now\_page = 1;  ViewModels.product\_homepage\_model.page.end\_page = (int)a;  }  #endregion  ViewModels.Products\_list = class1.Numberofpages<Production>(ViewModels.Products\_list, ViewModels.product\_homepage\_model.page.numbers, ViewModels.product\_homepage\_model.page.now\_page);    return View(ViewModels);  } | | **View** | | @using orders.Models.ViewModels  @model OD\_ViewModels  <style>  table.custom-table {  width: 100%; /\* 設定表格寬度為 80% \*/  height: 50px; /\* 設定表格高度為 200px \*/  border: 1px black;  }  th {  border: 1px solid black; /\* 表頭藍色邊框 \*/  padding: 10px;  }  td {  border: 1px solid black; /\* 單元格黑色邊框 \*/  padding: 10px;  }  </style>  <div class="jumbotron">  <h2>商品清單列</h2>  @using (Html.BeginForm("Product\_homepage\_Index", "orders", FormMethod.Post, new { @class = "form-horizontal", role = "form",id= "Form\_Product\_homepage" }))  {  // Html.BeginForm會將{}中宣告的物件回傳到Product\_homepage\_Index  @Html.HiddenFor(m => m.product\_homepage\_model.page.now\_page)  @Html.HiddenFor(m => m.product\_homepage\_model.page.end\_page)  @Html.HiddenFor(m => m.product\_homepage\_model.page.numbers)  <h>類型:</h> @Html.DropDownListFor(x => x.product\_homepage\_model.type\_id, Model.Types\_list, "請選擇")  <h>商品名稱:</h> @Html.TextBoxFor(x => x.product\_homepage\_model.product\_name)  <button>查詢</button>  }  </div>  @{  if (Model.Products\_list.Count > 0)  {  <div class="jumbotron">  <table class="custom-table">  <tr>  <th>  商品名稱  </th>  <th>  庫存數量  </th>  <th>  價格  </th>  </tr>  @foreach (var pd in Model.Products\_list)  {  <tr>  <th style="width:250px">  @pd.name  </th>  <th style="width:250px">  @pd.number  </th>  <th style="width:250px">  @pd.price  </th>  </tr>  }  <tr>  <th colspan="3">  <div class="ts-pagination is-fluid">  <a class="item is-back" onclick="changePage('up')">上一頁</a>  @for (int a = 1; a <= Model.product\_homepage\_model.page.end\_page; a++)  {  if (a < 5)  {  <a class="item" onclick="changePage(@a)"> @a </a>  }  else { break; }  }  <a class="item is-next" onclick="changePage('down')">下一頁</a>  </div>  </th>  </tr>  </table>  </div>  }  }  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>  <script>  function changePage(page) {  var now\_page = $('#product\_homepage\_model\_page\_now\_page').val();  var end\_page = $('#product\_homepage\_model\_page\_end\_page').val();  if (!isNaN(Number(page))) {  now\_page = page;  } else if (page == 'up') {  now\_page = Number(now\_page) - 1;  } else if (page == 'down') {  now\_page = Number(now\_page) + 1;  }  if (now\_page >= 1 && now\_page <= end\_page) {  $('#product\_homepage\_model\_page\_now\_page').val(now\_page);  //將畫面上的FORM在宣告後，要求執行此FORM  var from = document.getElementById("Form\_Product\_homepage");  from.submit();  }  }  </script> | |