|  |
| --- |
| 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;  } | |

|  |
| --- |
| Model |
| |  | | --- | | **變數宣告** | | public class ProductionDetail  {  public string productiondetail\_id { get; set; }//自動增長的編號  public string inout\_id { get; set; }//進出貨  public string product\_id { 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; }//創建日期  public DateTime? dt\_date { get; set; }//更新日期  }  public class Production\_model  {  public string type\_id { get; set; }//類型編號  public string product\_name { get; set; } //產品名稱  public string product\_id { get; set; } //產品編號  public List<SelectListItem> select\_type\_id { get; set; }  public List<SelectListItem> select\_inout\_id { get; set; }  public List<SelectListItem> select\_product { get; set; }  public page pd\_page { get; set; }//頁數  } | | **呼叫SQL預設資料設定** | | public Production\_model Set\_Production\_model()  {  Production\_model return\_value = new Production\_model();  return\_value.select\_type\_id = Types\_list();  return\_value.select\_product = ALLProduct\_list();  return\_value.select\_inout\_id = new List<SelectListItem>();  return\_value.select\_inout\_id.Add(new SelectListItem { Text = "入庫", Value = "0" });  return\_value.select\_inout\_id.Add(new SelectListItem { Text = "報銷", Value = "1" });  return return\_value;  } | | public List<SelectListItem> Types\_list()  {  string sql = "SELECT type\_id AS Value, name AS Text FROM API.dbo.Types WITH(NOLOCK)";  return class1.DataTableToListWithJson<SelectListItem>(class1.SQL\_select\_exce(sql));  }  public List<SelectListItem> ALLProduct\_list()  {  string sql = $@"SELECT product\_id AS Value, name AS Text FROM orders.dbo.Production WITH(NOLOCK)";  return class1.DataTableToListWithJson<SelectListItem>(class1.SQL\_select\_exce(sql));  } | | **呼叫SQL 資料清單** | | public List<Production> Production\_list(string type\_id, string product\_name,string product\_id)  {  List<Production> return\_list = new List<Production>();  string sql\_list = string.Empty;  if (!String.IsNullOrEmpty(type\_id))//有資料  {  sql\_list += $"and p.type\_id='{type\_id}'";  }  if (!String.IsNullOrEmpty(product\_id))//有資料  {  sql\_list += $"and p.product\_id='{product\_id}'";  }  if (!String.IsNullOrEmpty(product\_name))//有資料  {  sql\_list += $"and p.name='{product\_name}'";  }  sql\_list = $@"select t.name as type\_name,p.\* from orders.dbo.Production as p with (nolock)  left join orders.dbo.Types as t with (nolock) on p.type\_id = t.type\_id  where 1=1 {sql\_list}  order by p.number";  // 資料讀取  return\_list = class1.DataTableToListWithJson<Production>(class1.SQL\_select\_exce(sql\_list));  return return\_list;  } | | **呼叫SQL資料新增修改語法** | | public void Insert\_update\_Products(List<Production> input\_list)  {  string sql\_list = string.Empty;  try{  foreach (var product in input\_list)  {  if (product.type\_id != null && product.name != null && product.price != 0)  {  sql\_list = $@"  DECLARE @NewID INT;  SET @NewID = (SELECT ISNULL(MAX(CAST(SUBSTRING(product\_id, 2, 4) AS INT)), 0)  FROM API.dbo.Production);  IF EXISTS(select 1 from API.dbo.Production where product\_id='{product.product\_id ?? ""}')  BEGIN  update API.dbo.Production  Set  type\_id=TRIM('{product.type\_id ?? ""}'),  name=TRIM('{product.name ?? ""}'),  number={product.number},  price={product.price},  dt\_date=SYSDATETIME()  where product\_id='{product.product\_id ?? ""}'  END  ELSE  BEGIN  insert into API.dbo.Production  (  product\_id,  type\_id,  name,  number,  price,  dr\_date  )  values  (  'P' + RIGHT('000' + CAST(@NewID + 1 AS VARCHAR(4)), 4), -- 生成唯一的 OrderDetail ID  TRIM('{product.type\_id ?? ""}'),  TRIM('{product.name ?? ""}'),  {product.number},  {product.price},  SYSDATETIME()  )  END";  // 將 SqlDataReader 轉換為 List<Product>  class1.SQL\_select\_exce(sql\_list);  }  }  }  catch(Exception ex){}  } | |

|  |
| --- |
| Controller |
| |  | | --- | | **Production\_Index 內碼 商品新增修改刪除** | | public ActionResult Production\_Index()  {  OD\_ViewModels ViewModels = new OD\_ViewModels();  OD\_Models Models = new OD\_Models();  ViewModels.production\_model = Models.Set\_Production\_model();  ViewModels.Production\_list = Models.Production\_list(null,null,null);  for (int a = 0; a < 5; a++)  {  ViewModels.Production\_list.Add(new Production());  }  return View(ViewModels);  }  [HttpPost]  public ActionResult Production\_Index(OD\_ViewModels aj\_s)  {  OD\_ViewModels ViewModels = new OD\_ViewModels();  OD\_Models Models = new OD\_Models();  ViewModels.production\_model = Models.Set\_Production\_model();  ViewModels.production\_model.type\_id = aj\_s.production\_model.type\_id??null;  ViewModels.production\_model.product\_id = aj\_s.production\_model.product\_id ?? null;  ViewModels.production\_model.product\_name = aj\_s.production\_model.product\_name ?? null;  ViewModels.Production\_list = Models.Production\_list(aj\_s.production\_model.type\_id, aj\_s.production\_model.product\_name,  aj\_s.production\_model.product\_id);  for (int a = 0; a < 5; a++)  {  ViewModels.Production\_list.Add(new Production());  }  return View(ViewModels);  } | | **IS\_UP\_Production 資料新增修改** | | [HttpPost]  public ActionResult IS\_UP\_Production(OD\_ViewModels aj\_s)  {  OD\_ViewModels ViewModels = aj\_s;  OD\_Models Models = new OD\_Models();  try  {  Models.Insert\_update\_Products2(aj\_s.Production\_list);  }  catch (Exception ex){}  return RedirectToAction("Production\_Index");  } | |

|  |
| --- |
| View |
| |  | | --- | | **ProductionDetail\_history\_Index.cshtml** | | @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>庫存數量清單</h2>  @using (Html.BeginForm("Production\_Index", "orders", FormMethod.Post, new { @class = "form-horizontal", role = "form" }))  {  <h>類型:</h> @Html.DropDownListFor(x => x.production\_model.type\_id, Model.production\_model.select\_type\_id, "請選擇一項")  <h>商品編號:</h> @Html.TextBoxFor(x => x.production\_model.product\_id)  <h>商品名稱:</h> @Html.TextBoxFor(x => x.production\_model.product\_name)  <button>查詢</button>  }  </div>  @Html.ActionLink("入庫/報銷", "ProductionDetail\_Index", "orders", null, new { @class = "btn btn-primary", id = "WareReim" })  <button id="update" class="btn btn-primary" onclick="toggleEdit(@Model.Production\_list.Count)">新增或修改</button>  @using (Html.BeginForm("IS\_UP\_Production", "orders", FormMethod.Post))  {  <button id = "inser\_update" class = "btn btn-primary" style = "display: none; width: 100%">確認新增修改</button>  <br />  <br />  <h>資料共:</h> @(Model.Production\_list.Count-5)  <div class="jumbotron">  <h4>\*為必填</h4>  <table class="custom-table">  <tr>  <th>  類別 \*  </th>  <th>  商品編號  </th>  <th>  商品名稱 \*  </th>  <th>  庫存數量 \*  </th>  <th>  銷售單價 \*  </th>  </tr>  @for (int a = 0; a < Model.Production\_list.Count; a++)  {  <tr>  <th>  @Html.DropDownListFor(m => m.Production\_list[a].type\_id, Model.production\_model.select\_type\_id, "請選擇"  , new { @style = "width: 100px;border: none;", @disabled = "disabled" })  @Model.Production\_list[a].type\_name  </th>  <th>  @Html.TextBoxFor(m => m.Production\_list[a].product\_id, new { @style = "width: 100px;border: none;",  @readonly = "readonly" })  </th>  <th>  @Html.TextBoxFor(m => m.Production\_list[a].name, new { @style = "width: 100px;border: none;" ,  @readonly = "readonly" })  </th>  <th>  @Html.TextBoxFor(m => m.Production\_list[a].number, new { @style = "width: 100px;border: none;" ,  @readonly = "readonly" })  </th>  <th>  @Html.TextBoxFor(m => m.Production\_list[a].price, new { @style = "width: 100px;border: none;" ,  @readonly = "readonly" })  </th>  </tr>  }  </table>  </div>  }  <script type="text/javascript">  function toggleEdit(index) {  for (let a = 0; a < index; a++) {  var WareReim = document.getElementById("WareReim")  var update = document.getElementById("update");  var inser\_update = document.getElementById("inser\_update");  var DropDown1 = document.getElementById("Production\_list\_" + a + "\_\_type\_id");  var textbox2 = document.getElementById("Production\_list\_" + a + "\_\_product\_id");  var textbox3 = document.getElementById("Production\_list\_" + a + "\_\_name");  var textbox4 = document.getElementById("Production\_list\_" + a + "\_\_number");  var textbox5 = document.getElementById("Production\_list\_" + a + "\_\_price");  inser\_update.style.display = "block"; // 顯示按鈕  update.style.display = "none"; // 隱藏按鈕  WareReim.style.display = "none"; // 隱藏按鈕  if (textbox2 == null) {  // 切換 readonly 屬性  DropDown1.disabled = !DropDown1.disabled;  textbox2.readOnly = !textbox2.readOnly;  textbox3.readOnly = !textbox3.readOnly;  textbox4.readOnly = !textbox4.readOnly;  textbox5.readOnly = !textbox5.readOnly;  } else {  // 切換 readonly 屬性  DropDown1.disabled = !DropDown1.disabled;  textbox3.readOnly = !textbox3.readOnly;  textbox5.readOnly = !textbox5.readOnly;  }  }  }  </script> | |