|  |
| --- |
| SQL 預設程式建立 |
| |  | | --- | | **建立** | |  | | **SQL 預設程式** | | USE [orders]  GO  /\*\*\*\*\*\* Object: StoredProcedure [dbo].[InsertOrdersData] Script Date: 2024/12/7 下午 02:48:24 \*\*\*\*\*\*/  SET ANSI\_NULLS ON  GO  SET QUOTED\_IDENTIFIER ON  GO  ALTER PROCEDURE [dbo].[InsertOrdersData]  @Orders dbo.OrderDetailType2 READONLY,  @customer\_name VARCHAR(50),  @factory\_name VARCHAR(50)  AS  BEGIN  -- 宣告變數  DECLARE @NewOrderID VARCHAR(10);  DECLARE @NewOrderDetailID VARCHAR(10);  -- 檢查庫存和產品是否符合條件  IF((SELECT COUNT(\*) FROM orders.dbo.Production AS p  LEFT JOIN @Orders AS o ON p.product\_id = o.product\_id  WHERE p.number > o.number AND p.product\_id = o.product\_id)<>  (SELECT COUNT(\*) FROM @Orders))  BEGIN  -- 如果條件不符，返回錯誤  RAISERROR('下單不成功，請重新下單', 16, 1);  RETURN;  END  ELSE  BEGIN  -- 更新庫存數量  UPDATE p  SET p.number = p.number - od.number  FROM @Orders AS od  LEFT JOIN orders.dbo.Production AS p ON od.product\_id = p.product\_id;  -- 生成新訂單號  SELECT @NewOrderID = 'O' + RIGHT('0000' + CAST(ISNULL(CAST(SUBSTRING(MAX(order\_id), 3, 4) AS INT), 0) + 1 AS VARCHAR(4)), 4)  FROM orders.dbo.Orders;    -- 插入新訂單  INSERT INTO orders.dbo.Orders (order\_id, name, factory\_name, dr\_date)  VALUES (  @NewOrderID,  @customer\_name,  @factory\_name,  SYSDATETIME()  );  SET @NewOrderDetailID = ISNULL((SELECT MAX(CAST(SUBSTRING(orderdetail\_id, 3, 4) AS INT)) FROM orders.dbo.OrderDetail), 0);  -- 逐筆插入訂單詳情  INSERT INTO orders.dbo.OrderDetail (orderdetail\_id, order\_id, product\_id, number, price, dr\_date)  SELECT  'OD' + RIGHT('0000' + CAST(ROW\_NUMBER() OVER(ORDER BY (SELECT @NewOrderDetailID)) + @NewOrderDetailID AS VARCHAR(4)), 4), -- 生成唯一的 OrderDetail ID  @NewOrderID, -- 關聯新訂單ID  o.product\_id,  o.number,  o.price,  SYSDATETIME()  FROM @Orders o;  RAISERROR('下單成功', 16, 1);  RETURN;  END  END; | |

|  |
| --- |
| 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FUNCTION** | | public string SQL\_exce\_spectail(string sqlstring, DataTable data, string name, string factory\_name)  {  string connectionString = "連資料庫路徑";  SqlDataReader reader;  string datas = "";  try  {  using (SqlConnection conn = new SqlConnection(connectionString))  {  // 使用參數化查詢呼叫預存程序  using (SqlCommand cmd = new SqlCommand("[orders].dbo.[InsertOrdersData]", conn))  {  // 指定使用預存程序  cmd.CommandType = CommandType.StoredProcedure;  // 設定表值參數  SqlParameter ordersParam = cmd.Parameters.AddWithValue("@Orders", data);  ordersParam.SqlDbType = SqlDbType.Structured; // 指定為表值參數  ordersParam.TypeName = "dbo.OrderDetailType2"; // 指定表值參數的型別  // 設定其他參數  cmd.Parameters.AddWithValue("@customer\_name", name); // 假設值 '1'  cmd.Parameters.AddWithValue("@factory\_name", factory\_name); // 假設值 '2'  // 打開連接  conn.Open();  // 執行命令  reader = cmd.ExecuteReader();  }  }  }  catch (Exception ex)  {  datas = ex.Message;  }  return datas;  } | |

|  |
| --- |
| MODEL |
| |  | | --- | | **變數宣告** | | public class New\_Order  {  public string name { get; set; }//顧客  public string factory\_name { get; set; }//工廠編號  public string Message { get; set; }//顧客  public DateTime? arrival\_date { get; set; }//到貨日  public DateTime? dr\_date { get; set; }  public DateTime? dt\_date { get; set; }  public List<SelectListItem> select\_factory { get; set; }  public List<SelectListItem> select\_customername { get; set; }  public List<OrderDetial\_list> OrderDetial\_list { get; set; }  public List<SelectListItem> select\_product { get; set; }  }  public class OrderDetial\_list  {  public string product\_id { get; set; }//編號  public int number { get; set; }//銷售數量  public int Productionnumber { get; set; }//銷售數量  public string price { get; set; } //產品售價  public DateTime? dr\_date { get; set; }  public DateTime? dt\_date { get; set; }  } | | **呼叫 預設設定** | | public New\_Order Set\_New\_Order()  {  New\_Order return\_value = new New\_Order();  return\_value.select\_factory = Factory\_list();  return\_value.select\_customername = CustomerName\_list();  return\_value.select\_product = Product\_list();  return return\_value;  } | | // 工廠下拉選單  public List<SelectListItem> Factory\_list()  {  string sql = "SELECT factory\_id AS Value, name AS Text FROM orders.dbo.Factory WITH(NOLOCK)";  return Class1.DataTableToListWithJson<SelectListItem>(class1.SQL\_select\_exce(sql));  }  // 客戶名稱下拉選單  public List<SelectListItem> CustomerName\_list()  {  string sql = "SELECT customer\_id AS Value, name AS Text FROM orders.dbo.Customer WITH(NOLOCK)";  return Class1.DataTableToListWithJson<SelectListItem>(class1.SQL\_select\_exce(sql));  }  // 產品下拉選單  public List<SelectListItem> Product\_list()  {  string sql = $@"SELECT product\_id AS Value, name AS Text FROM orders.dbo.Production WITH(NOLOCK)  where number> 0";  return Class1.DataTableToListWithJson<SelectListItem>(class1.SQL\_select\_exce(sql));  } | | **取的商品清單 (需要商品庫存/單價)** | | 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;  } | | **呼叫使用SQL 預存程式處存資料** | | public string EXCE\_OrderDetail(New\_Order OrderDetail)  {  string return\_list = "";  string sql\_list = string.Empty;  DataTable data = new DataTable();  data.Columns.Add("product\_id", typeof(string));  data.Columns.Add("number", typeof(int));  data.Columns.Add("price", typeof(int));  foreach (var alls in OrderDetail.OrderDetial\_list)  {  if (alls.product\_id != null)  {  DataRow row = data.NewRow();  row["product\_id"] = alls.product\_id;  row["number"] = alls.number;  row["price"] = alls.price;  data.Rows.Add(row);  }  }  if (OrderDetail.name != null && OrderDetail.factory\_name != null)  {  try  {  //EXEC InsertOrdersProcedure @Orders = @orderDetails, @name = 'John Doe', @price = 30.50;  sql\_list = $@"exec [API].dbo.[InsertOrdersData]  @Orders=@OrdersDate,  @customer\_name='{OrderDetail.name}',  @factory\_name='{OrderDetail.factory\_name}'";  return\_list= class1.SQL\_exce\_spectail(sql\_list, data, OrderDetail.name,OrderDetail.factory\_name);  } catch (Exception ex)  {  return\_list = ex.Message;  }  }  else  {  return\_list = "值有錯誤";  }  return return\_list;  } | |

|  |
| --- |
| Contoller |
| |  | | --- | | **Orderdetils\_Index 內碼** | | public ActionResult Orderdetils\_Index()  {  OD\_ViewModels ViewModels = new OD\_ViewModels();  OD\_Models Models = new OD\_Models();  ViewModels.New\_Order\_model = Models.Set\_New\_Order();  ViewModels.New\_Order\_model.OrderDetial\_list = new List<OrderDetial\_list>();  for (int a = 0; a < 9; a++)  {  ViewModels.New\_Order\_model.OrderDetial\_list.Add(new OrderDetial\_list());  }  return View(ViewModels);  }  [HttpPost]  public ActionResult Orderdetils\_Index(List<OrderDetial\_list> ajs)  {  OD\_ViewModels ViewModels = new OD\_ViewModels();  OD\_Models Models = new OD\_Models();  List<OrderDetial\_list> return\_list = new List<OrderDetial\_list>();  ViewModels.Production\_list = Models.Production\_list(null, null,null);  for (int a = 0; a < ajs.Count; a++)  {  return\_list.Add(ajs[a]);  if (ajs[a].product\_id != null)  {  return\_list[a].Productionnumber = ViewModels.Production\_list.Find(x => x.product\_id == return\_list[a].product\_id).number;  return\_list[a].price = ViewModels.Production\_list.Find(x => x.product\_id == return\_list[a].product\_id).price.ToString();  }  }  return Json(return\_list);  } | | **呼叫SQL資料新增並顯示是否成功** | | public ActionResult Orderdetils\_check(OD\_ViewModels ajs)  {  OD\_ViewModels ViewModels = new OD\_ViewModels();  OD\_Models Models = new OD\_Models();  try{  ViewModels.New\_Order\_model = Models.Set\_New\_Order();  ViewModels.Production\_list = Models.Production\_list(null, null,null);  ViewModels.New\_Order\_model.OrderDetial\_list = ajs.New\_Order\_model.OrderDetial\_list;  ViewModels.New\_Order\_model.Message= Models.EXCE\_OrderDetail(ajs.New\_Order\_model);  ViewModels.New\_Order\_model.name = ajs.New\_Order\_model.name;  ViewModels.New\_Order\_model.factory\_name = ajs.New\_Order\_model.factory\_name;  }  catch (Exception ex)  {  return RedirectToAction("Orderdetils\_Index");  }  return View(ViewModels);  } | |

|  |
| --- |
| View |
| |  | | --- | | **Orderdetils\_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;  }  table.title-table {  width: 100%; /\* 設定表格寬度為 80% \*/  height: 50px; /\* 設定表格高度為 200px \*/  }  td {  border: 1px solid black; /\* 單元格黑色邊框 \*/  padding: 10px;  }  th {  /\*border: 1px solid black;\*/ /\* 單元格黑色邊框 \*/  /\*height: 50px;\*/ /\* 設定表格高度為 200px \*/  text-align: center;  }  </style>  <div class="jumbotron">  <h3>商品訂貨</h3>  <div class="progress">  <div id="progressbar" class="progress-bar bg-success" role="progressbar" style="width: 50%" aria-valuenow="25" aria-valuemin="0" aria-valuemax="100"></div>  </div>  <table class="title-table">  <tr>  <th>訂單資料填寫</th>  <th>確認是否下單</th>  </tr>  </table>  </div>  <div class="jumbotron">  <h4><button class="btn btn-default" onclick="test()" id="getvalue\_btn">帶出資料</button></h4>  <h4><button class="btn btn-default" onclick="update()" id="update\_btn" style="display:none;">重新填寫</button></h4>  @using (Html.BeginForm("Orderdetils\_check", "orders", FormMethod.Get, new { onsubmit = "return input\_value\_check('end')" }))  {  <h4><button class="btn btn-default" style="display:none;" id="next\_btn">確認送出</button></h4>  <h4>工廠名稱 \*： @Html.DropDownListFor(m => m.New\_Order\_model.factory\_name, Model.New\_Order\_model.select\_factory,  "請選擇",new { @style= "width: 150px;" }) </h4>  <h4>顧客名稱 \*： @Html.DropDownListFor(m => m.New\_Order\_model.name, Model.New\_Order\_model.select\_customername,  "請選擇" ,new { @style = "width: 150px;" }) </h4>  <table class="custom-table">  <tr>  <td style="text-align: center;">編號</td>  <td style="text-align: center;">產品\*</td>  <td style="text-align: center;">數量\*</td>  <td style="text-align: center;">單價</td>  <td style="text-align: center;">庫存數量</td>  </tr>  @for (int a = 0; a < Model.New\_Order\_model.OrderDetial\_list.Count; a++)  {  <tr>  <td>  @(a+1)  </td>  <td>  <h4>@Html.DropDownListFor(m => m.New\_Order\_model.OrderDetial\_list[a].product\_id,  Model.New\_Order\_model.select\_product, "請選擇", new { @style = "width: 150px;border: none;",  @class = "form-control dropdown-check" })</h4>  </td>  <td>  <h4>@Html.TextBoxFor(m => m.New\_Order\_model.OrderDetial\_list[a].number)</h4>  </td>  <td>  <h4>@Html.TextBoxFor(m => m.New\_Order\_model.OrderDetial\_list[a].price,  new { @readonly = "readonly" })</h4>  </td>  <td>  <h4>@Html.TextBoxFor(m => m.New\_Order\_model.OrderDetial\_list[a].Productionnumber,  new { @readonly = "readonly" })</h4>  </td>  </tr>  }  </table>  }  </div>  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>  <script>  //為每個下拉選單添加監聽事件  $('.dropdown-check').on('click', function (event) {  if (isAllowedToOpen()) {  alert("你不能打開這個下拉選單！");  event.preventDefault(); // 阻止下拉選單被打開  }  });  // 判斷條件的函數，根據你的業務邏輯返回 true 或 false  function isAllowedToOpen() {  var someCondition = document.getElementById('getvalue\_btn').style.display == "none";  return someCondition; // 返回 true 表示允許，false 表示不允許  }  //資料必填判斷  function input\_value\_check(value) {  if ($('#New\_Order\_model\_name').val() == '' || $('#New\_Order\_model\_factory\_name').val() == '') {  alert('請填寫 顧客名稱 和 工廠名稱'); // 顯示提示信息  return false;  }  else {  let product\_id = '';  let number = '';  let Productionnumber = '';  let pt = '0';//1是TRUE，0是FALSE  for (let a = 0; a < @Model.New\_Order\_model.OrderDetial\_list.Count; a++) {  product\_id = '#New\_Order\_model\_OrderDetial\_list\_' + a + '\_\_product\_id';  number = '#New\_Order\_model\_OrderDetial\_list\_' + a + '\_\_number';  Productionnumber = '#New\_Order\_model\_OrderDetial\_list\_' + a + '\_\_Productionnumber';  if ($(product\_id).val() != '') {  pt = '1';  if (isNaN(Number($(number).val())) == true) {//如若值為空白或不為數字  alert('數量欄位請填寫數字'); // 顯示提示信息  return false;  } else {  var number\_s = Number($(number).val());  var Productionnumber\_s = Number($(Productionnumber).val());  console.log(number\_s < Productionnumber\_s)  if (number\_s==0) {  alert('請填寫 訂購商品數量'); // 顯示提示信息  return false;  } else if (value == 'end' && number\_s > Productionnumber\_s) {  console.log(value)  alert('訂購商品數量已經超過庫存量 需重新填寫'); // 顯示提示信息  return false;  }  }  }  }  //無填寫任何資料  if (pt == '0') {  alert('請填寫 訂購清單'); // 顯示提示信息  return false;  }  }  return true;  }  //顯示裝置卡控  function update() {  var progressBar = document.querySelector('.progress-bar');  //var $progressBar = $('#progressBar');  if (document.getElementById("getvalue\_btn").style.display == "none") {  document.getElementById('getvalue\_btn').style.display = "block";  document.getElementById('update\_btn').style.display = "none";  document.getElementById('next\_btn').style.display = "none";  let price = '';  let Productionnumber = '';  for (let a = 0; a < @Model.New\_Order\_model.OrderDetial\_list.Count; a++) {  price = '#New\_Order\_model\_OrderDetial\_list\_' + a + '\_\_price';  Productionnumber = '#New\_Order\_model\_OrderDetial\_list\_' + a + '\_\_Productionnumber';  $(price).val('')  $(Productionnumber).val(0)  }  // 设置新的 width 样式  progressBar.style.width = '50%';  } else {  document.getElementById('getvalue\_btn').style.display = "none";  document.getElementById('update\_btn').style.display = "block";  document.getElementById('next\_btn').style.display = "block";  // 设置新的 width 样式  progressBar.style.width = '100%';  }  }  //資料抓取  function test() {  let pt = '0';//1是TRUE，0是FALSE  let number = '';  let product\_id = '';  let price = '';  let Productionnumber = '';  var modelData = [];  if (input\_value\_check('start') == true) {  for (let a = 0; a < @Model.New\_Order\_model.OrderDetial\_list.Count; a++) {  number = '#New\_Order\_model\_OrderDetial\_list\_' + a + '\_\_number';  product\_id = '#New\_Order\_model\_OrderDetial\_list\_' + a + '\_\_product\_id';  var order = {  Productionnumber:'',  price:'',  product\_id: $(product\_id).val(),  number: $(number).val()  };  modelData.push(order);  }  $.ajax({  url: '@Url.Action("Orderdetils\_Index", "orders")',  type: 'POST',  contentType: 'application/json',  data: JSON.stringify(modelData), // 將資料轉換為 JSON  success: function (response) {  let a = 0;  pt = '1';  response.forEach(function (response) {  //賦予值  price = '#New\_Order\_model\_OrderDetial\_list\_' + a + '\_\_price';  Productionnumber = '#New\_Order\_model\_OrderDetial\_list\_' + a + '\_\_Productionnumber';  $(price).val(response.price)  $(Productionnumber).val(response.Productionnumber)  //判斷下單商品數量是否大於庫存數量  if (response.number > response.Productionnumber) {  pt = '0';  }  a++;  });  //下單商品數量小於/等於庫存數量  if (pt == "1") {  update();  } else {  alert('訂購商品數量已經超過庫存量 需重新填寫');  }  },  error: function () {  alert('Error occurred!');  }  });  }  }  </script> | | **Orderdetils\_check.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;  }  table.title-table {  width: 100%; /\* 設定表格寬度為 80% \*/  height: 50px; /\* 設定表格高度為 200px \*/  }  td {  border: 1px solid black; /\* 單元格黑色邊框 \*/  padding: 10px;  }  th {  width: 25%;  text-align: center;  }  </style>  <div class="jumbotron">  <h3>商品訂貨</h3>  <div class="progress">  <div class="progress-bar bg-success" role="progressbar" style="width: 100%" aria-valuenow="25" aria-valuemin="0" aria-valuemax="100"></div>  </div>  <table class="title-table">  <tr>  <th><h2> @Model.New\_Order\_model.Message</h2></th>  </tr>  </table>  </div>  <div class="jumbotron">  <h4>顧客名稱： @Html.DropDownListFor(m => m.New\_Order\_model.name, Model.New\_Order\_model.select\_customername, "請選擇", new { @style = "width: 150px;border: none;", @disabled = "disabled" }) </h4>  <h4>工廠名稱： @Html.DropDownListFor(m => m.New\_Order\_model.factory\_name, Model.New\_Order\_model.select\_factory, "請選擇", new { @style = "width: 150px;border: none;", @disabled = "disabled" }) </h4>  <table class="custom-table">  <tr>  <td>編號</td>  <td>產品</td>  <td>數量</td>  <td>價格</td>  <td>庫存數量</td>  </tr>  @for (int a = 0; a < Model.New\_Order\_model.OrderDetial\_list.Count; a++)  {  if (Model.New\_Order\_model.OrderDetial\_list[a].product\_id !=null)  {  <tr>  <td>  @(a+1)  </td>  <td>  <h4>@Html.DropDownListFor(m => m.New\_Order\_model.OrderDetial\_list[a].product\_id, Model.New\_Order\_model.select\_product, "請選擇", new { @style = "width: 150px;border: none;", @disabled = "disabled" })</h4>  </td>  <td>  <h4>@Model.New\_Order\_model.OrderDetial\_list[a].number</h4>  </td>  <td>  <h4>@Model.New\_Order\_model.OrderDetial\_list[a].price</h4>  </td>  <td>  <h4>@Model.New\_Order\_model.OrderDetial\_list[a].Productionnumber</h4>  </td>  </tr>  }  }  </table>  </div> | |