新增資料到資料庫 (Lucas 筆記)

Framework versions

ASP.NET (active / Page) → (MVC) / (Web Form)

建議好、新的不推薦使用舊的

ASP.NET MVC
- Model
- View
- Controller
- Route

User —ask a request→ server (Microsoft: ASP, ASP.NET, ASP.NET MVC; PHP, Java) —response a web→ User

User: Html, CSS, JavaScript

server ⇄ data: SQL server, mysql

Web Form: 建立專案 → ASP.NET Web 應用程式 (.NET Framework) → 空白 → 建立

MVC: 建立專案 → ASP.NET Web 應用程式 (.NET Framework) → MVC → 建立

pure → 調用 → 資料庫 → 到瀏覽器

推薦: 記名稱法

建立專案: 建立專案 → ASP.NET Web 應用程式 (.NET Framework) → 空白 → 在 MVC → 建立

cshtml @ → 專門寫給 C# 的檔案

@RenderBody() 主頭頁面的內容

in C# var ? + 給予名稱的匿名物件 參數

HtmlUtility.HtmlEncode( )

跳脫編碼

跳脫三法則

Auto → 新增路徑

Route.Config.cs

url: {controller}/{action}/{apple} → 可選的參數

defaults: new { controller = "Home", action = "Index", apple = UrlParameter.Optional }

HomeController.cs

public string welcome(string name, int apple=1)

瀏覽 _Layout.cshtml 的 View not found 時, 新增資料夾、檢查 bootstrap、jQuery

CSS 檢查 (Content → 樣式 → CSS) → 修改內容

inline

空白的 → Controllers 右鍵 → 加入 → 控制器 → MVC 5 控制器

基礎空白主控器, 沒未來 _Layout 資料夾, 例如 気候, View

自訂 自訂檢查右鍵 → 加入 → 檢視 → MVC 5 → 改名 (名稱為 index)

編輯 → 右鍵 → 加入 → 全域專案

移好 空白 View\Shared_Layout.cshtml (Layout 檢視設定)

引用的 檢視 的 Layout 設定 空白 → 自訂的 設定

@ViewContext.RouteData.Values["controller"]; "action"

新增自己的 Layout → MVC 5 檢視的專案 (Razor)

討論區 不同 Controller

不同 Layout

在 index 之外

- 在檢視的語法 資料
  - ViewBag
  - ViewData 無 cast 在資料, 沒有 型別 的資料型
  - TempData 失效機制 一次 取完即失
  - Session
  - Cookie HttpCookie

Razor 語法

@Html.ActionLink("顯示文字", "action", "controller")

~/ Controller/Action

in <> in cshtml

ViewBag.apple = 5 @ViewBag.apple show 5

ViewData["banana"] = 6 @ ViewData["banana"] show 6

TempData["car"] = 7 @ TempData["car"] show 7

Session["orange"] = 8 @ Session["orange"] show 8

new cookie @ Request.Cookies["myApple"].Value

在右邊一個檢視頁面 第一個 快取到

如果想要保留 TempData.Keep("123");

宣告變數

@{ string var = "apple"; }

使用

 @ var

多行

@{
var
var weekDay = DateTime.Now.DayOfWeek
var
}

物件 = 屬性(特徵) + 方法(動作)

@Request.Url 專案網址 路徑
- File Path 專案路徑
- MapPath 實體檔案在伺服器
- RequestType Get? 現在 POST

if (IsPost)(bool) → 判斷是否為第 2 次回到畫面

簡版 MVC 新專案 連結 Bootstrap 要使用的版本

清理用不到的檔案

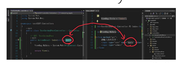
加入 控制器 (為 網頁 的多 view)

對應 index() 新增檢視

執行 調試 ↓ ▶ 2. 編譯右鍵 → 在瀏覽器中檢視

3. 畫面右鍵 → " ((Ctrl + Shift + V)

在 Controller 之 動作 - 利用 ViewBag 傳 資料 傳遞 至 View 第入 html. 用@ViewBag 引用資料自己在 加動作



在 index 傳 (string Apple) 收參數

連結到 Controller name 命名

直接用 ViewBag.Temp = Apple; 收參數 引用在 View 上顯示 ,

ViewBag.Temp.Value = Request["Apple"]; 無型別

MapPath 伺服器路徑 server = , BuaerryString("Apple") 上傳文件的資訊

專案內的 Global.asax 如果有 - 開始和結束的資料、然 調試 - 發生錯誤

任何調度新的錯誤 到 session 的值

Razor 語法

. IsInt()判斷 與 不能 型態

(轉型 . AsInt()

無 的 時間

@DateTime.Now

能夠文字解出 加 @: EX @if (1==1) {@: Hello }

@{ string content = "Cat"}

 @content 顯示 Cat

@ Html.Raw (@ Content) 顯示 Cat

Razor 語法 語

在controllers 寫 function @function { public string Sayhi() { return "Hi"; } }

Html 下拉清單 <select> / <option>

選擇 selected

預設 selected 收集資料

Auto + shift 拖曳

Route.Config 的 語

?: 三元運算子

判斷? 的 null 的 類型 型別 bool? 的 引數 null

?. 條件運算 & A?.B?.Date

主要 null 的 條件 直

ViewBag.MyPath = System.Web.HttpContext.Current.Request.HttpMethod;

取得目前 請求 的 HTTP 資料 傳遞 的 方法 object get, post, head

html type = "submit" 的 button 類型 submit

<form method="get" action = 網址 , 目的

利用 get 將 傳輸 入 參數 進入 url 傳遞

傳遞時 post 不會 "

自訂義 action 動作 命名 比 新增

form 的 參數 的 型別 vs. 命名 的 view

對應 - 取 public ActionResult UserLogin(string Apple) { viewBag.Temp4 = Fluent.Apply(); return View; }

public class Mycar { public string Apple {get;set;} }

return View(viewBag.Temp4);

|結束 actionresult 回傳|

① return View();

② return View("viewcontroller/自訂名稱") 常見的 About 制模式 注意 這 action Result 不會 從 到 別的 "About"

③ return View("~/view / Home / About.cshtml") 可以跨 controller

④ return Content("xxx內容") 只回傳字串 xxx內容 用於 測試 也 加 html 自定 字串 語系 (text/plain) / (html) 檢查值

return Content("<h1>Apple</h1>","text/plain") <h1>Apple</h1>

⑤ return Redirect("網址")

⑥ return Redirect("網頁的 URL")

("~/Home/About")

("~/Test Action Result/About")

⑦ return RedirectToAction("action", "controller") 可以跨 controller

About Test Action Result

|引語 Jsonn|

加入 Using System.Web.Script.Serialization;

JavaScriptSerializer js = new JavaScriptSerializer();

string data = js.Serialize(products); > 將 List 轉成字串

List < > products;

$.ajax({ url: "", 接資料 success: function() { } 任務結束 }

JSON.parse( ) 分 將 JSON 型 物件

$().each(): 遍歷取得 元素 物

$.each(): 讀取 陣列 到 畫面 的 值 目 $.each(陣列名 object , function (idx, elm) idx = index

|使用 PartialView 設計 調用 部分 畫面 的 函數|

① 右鍵 → 新增 檢視 → 加入 → 選項: 回 產生 強型 檢視

return PartialView("action", 資料 list)

部分 view @model IEnumerable<

@foreach( )

部分 結束

- 注意 語法

② 加入 → MVC 5 部分 檢視 (Razor)

@Html.Partial("") AJ, Ajax

@Html.RenderAction("") Ajaxmany

產生 action

注意 的 前置 NEW, New, new 等 字串, 判斷 前 常 不 轉 大小寫

|_Layout 的 Razor|

@RenderBody 母 頭 section

BundleConfig.cs script.ui

bundles.Add(new StyleBundle("name").Include("~/ 檢視.css", site.css"));

View 引用

@Styles.Render("name")

@Scripts.Render("name")

@RenderSection("apple", required: false) 多 次 定義 命名 必 用 此 此

View

@section apple { <script> console.log(5); <script> } 總是 語句

引入 多個

|轉換 model|

右鍵 → 加入 → 新增項目 → 資料模型 → 實體 .cs

@foreach (var item in Model)

@Html.DisplayNameFor 顯示 欄位 名稱

@Html.DisplayFor (model item => item.StudentID) 顯示 理由 值 的 值

@Html.ActionLink("Create New", "Create")

get @Helper CheckScore(int score) { if (score<60) { <td class="red">@score</td> } else { <td>@score</td> } }

usef @CheckScore(item.Chinese);

|Model 檢查 建立|

Model 檢查器 2 • 使用 - mvc 使用 EntityFrameworkCore 搭配 Models 資料夾

引擎: 加入 脚架控制 • 資料: ASP.NET 脚架 資料類型 •

模型類別: Product • 資料類別: ProductContext •

資料提供程式: Microsoft.EntityFrameworkCore •

檢視: 從 Razor 產生檢視

控制器: 從 Razor 產生檢視

控制器名稱: ProductsController

<> product 產生 產生 產生