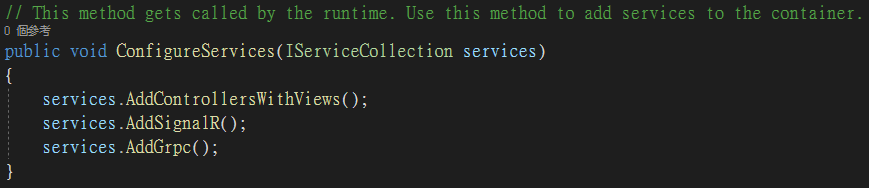
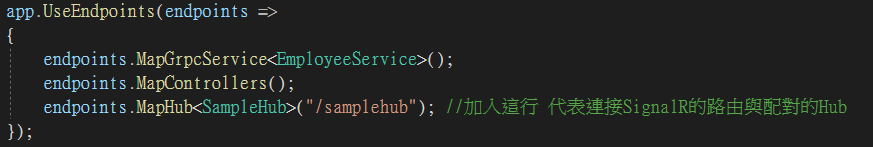
* 使用SignalR讓Client間可以同步資訊

於Startup.cs中->ConfigureService函式中加入SignalR實體並設置ControllerWithViews



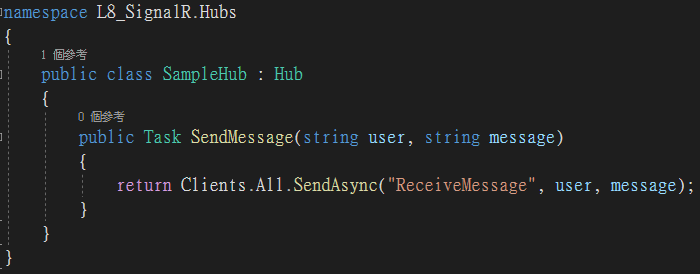
於Startup.cs中->Configure函式中UseEndpoints加入SignalR的Hub設定



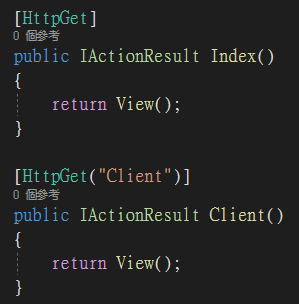
建立SignalR的Hub並設定其資料共享函式



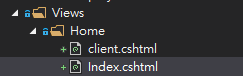
當Client使用SendMessage傳送資料到Hub而其他用戶則會由ReceiveMessage來接收訊息



創造一個Controller.cs並使用return View()方法來呈現頁面



建立View資料夾並且增加View相關檔案(index.cshtml)



首先必須載入SignalR.js函式庫 (<https://cdnjs.cloudflare.com/ajax/libs/microsoft-signalr/3.1.7/signalr.js>)

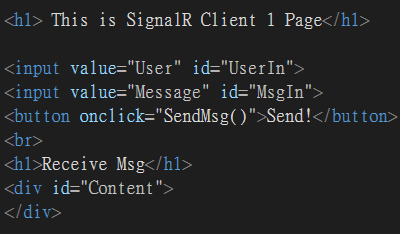
使用signalR.HubConnectionBuilder()連接我們所定義的sampleHub Server



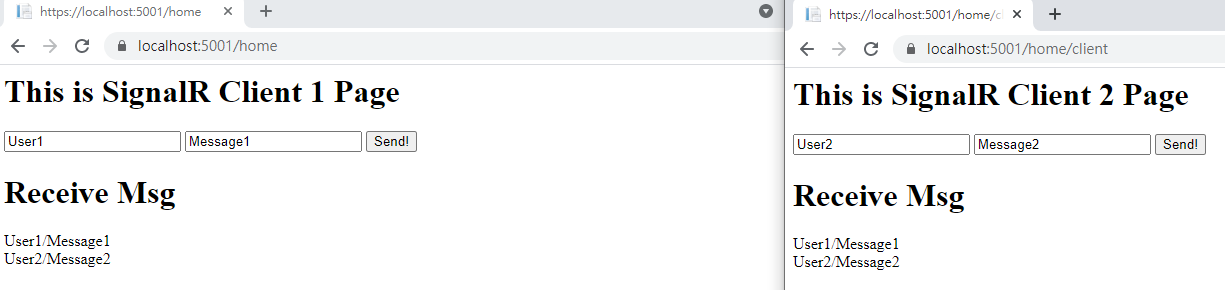
使用我們於SampleHub.cs所定義的SendMessage與ReceiveMessage方法來送收資料



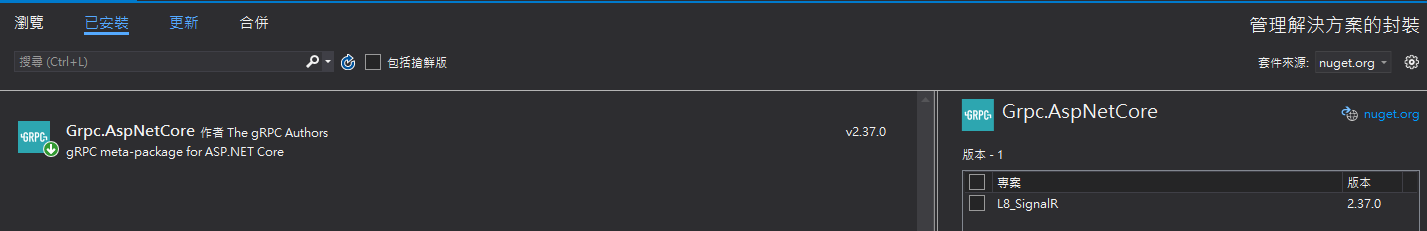
撰寫一些接收資料呈現的UI並進行測試



連入localhost/home與home/client就可以看到兩邊資訊同步更新



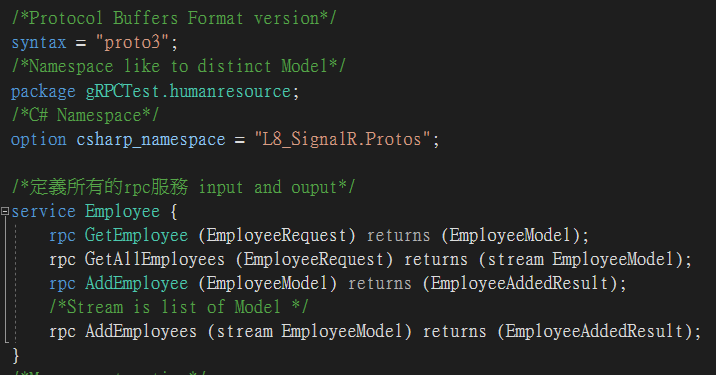
* gRPC Server建置

NuGet安裝Grpc.AspNetCore套件

新增Protos資料夾並新增Proto檔案



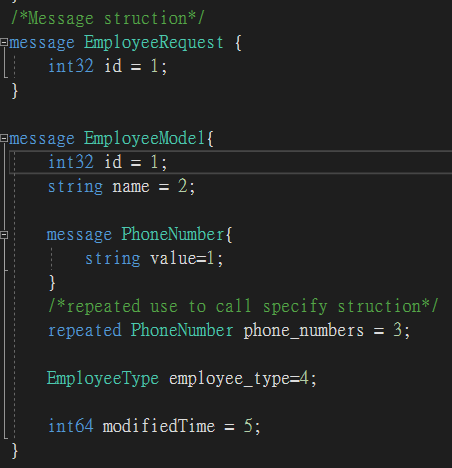
撰寫Proto檔案 (Proto檔案會定義服務的格式與使用方法)



package為你這個gRPC的名稱

option csharp\_namespace則為gRPC的namespace後面引用時使用這個名稱

使用 rpc [服務名稱] [input資料格式] return [output資料格式]來定義服務

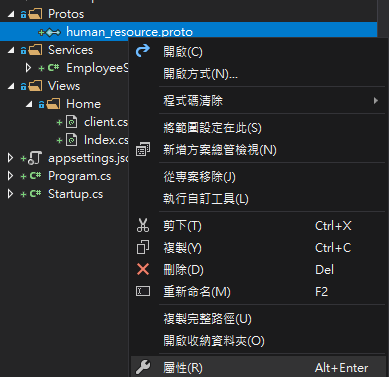


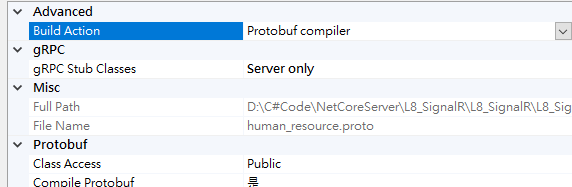
訂定資料格式，以message EmployeeModel為例

int32 id = 1 代表第一個欄位是int32格式名稱叫做id

string name = 2 代表第二個欄位是string格式名稱叫做name，後面以此類推

設定Protos檔案資訊，右鍵Proto檔案->屬性->Build Action選擇Protobuf compliler->gRPC Stub Classes選擇Server only

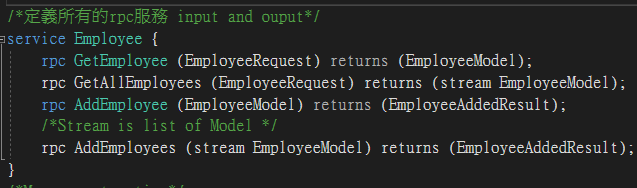


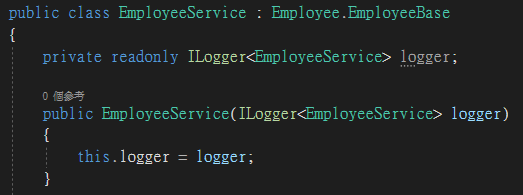


定義服務主體，先創立Services資料夾在裡面創建一個EmployeeService檔案

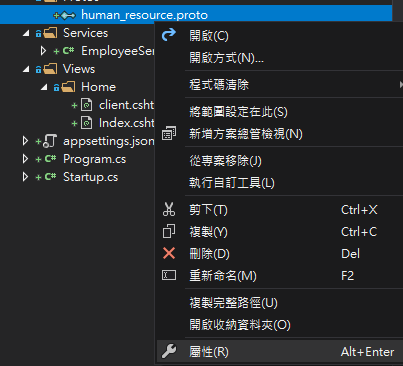


於EmployeeService創建一個Class，Class名稱會與Protos檔案定義的一樣

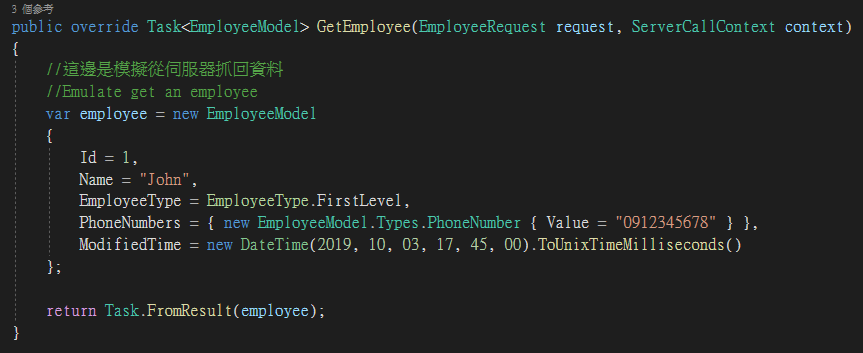


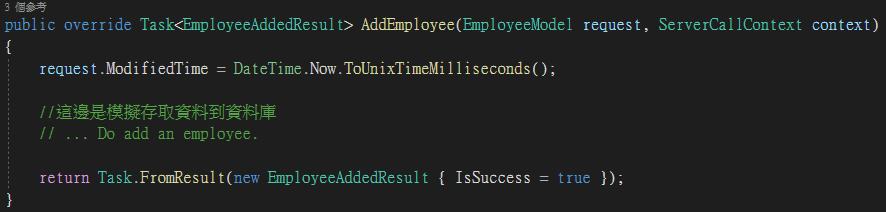


設定此Proto檔案為gRPC Server所使用的Proto檔案



於該Class當中定義你的服務運作主體，Input/Output需與Protos定義的一樣





於Startup.cs->Configure函式當中的app.UseEndpoints加入Grpc服務如下

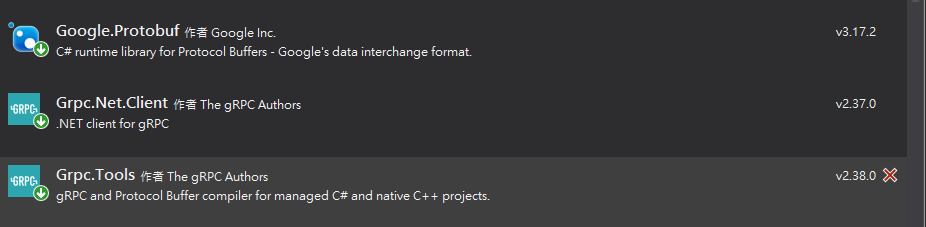


* gRPC Client建置

建立一個Console新專案命名為GrpcClient

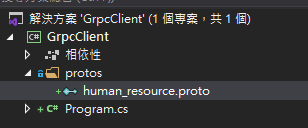
於NuGet當中加入下列三個套件

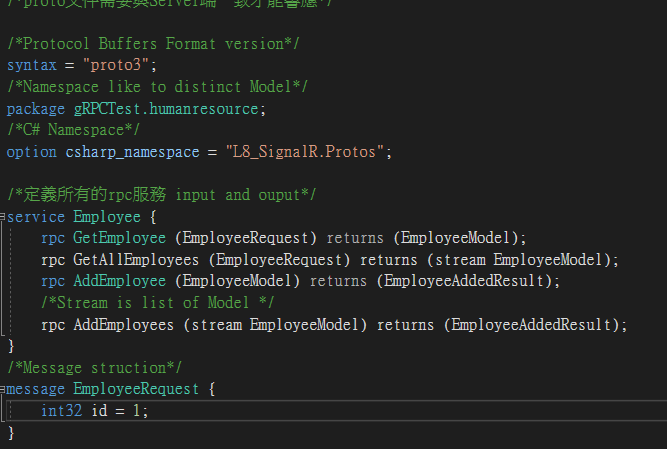
Google.Protobuf/Grpc.Net.Client/Grpc.Tools



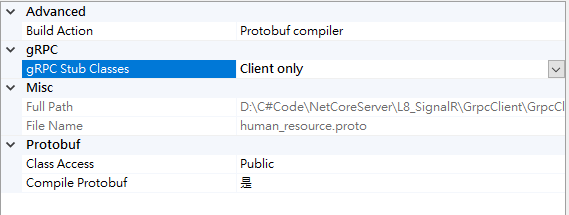
將Server的Protos資料加複製到GrpcClient專案中

須保證Server與Client使用的Proto檔案為同一份



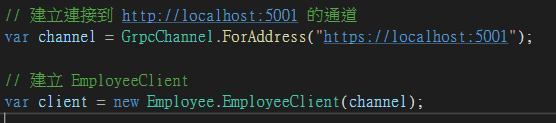


設置該Proto檔案屬性為Client only

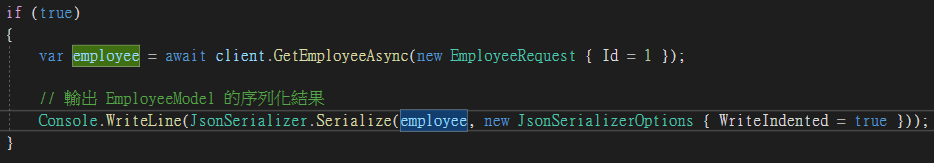


在Main Function當中->GrpcChannel.ForAddress([gRPC Server Ip:Port])連接Server

使用 Employee.EmployeeClient()來連接服務



可以試著使用client.GetEmployeeAsync()/GetAllEmployees()/AddEmployees()



先運行gRPC Server在運行gRPC Client就可以看到Server成功回應

