1. Mô hình thực hiện

Client

Client

API (HTTP)

Client

Client

Theo sơ đồ trên:

TH1:

Các client web connect với Hub, và API

Các client khác (Mobile) connect với API

Khi client mobile thực thi một phương thức trên API thì tất cả các client đều cập nhật mới

TH2: Mở rộng

Các client web connect với Hub, và API

Các client khác (Mobile) connect với API, với Hub

Khi bất kỳ 1 client nào gọi bất kỳ phương thức API nào thì tất cả các client khác đều nhận được thông báo.

TH3: (trường hợp tối giản cho trường hợp 3)

Các clients mobile và web chỉ cần kết nối tới Hub (có thể bỏ các phương thức API, viết sẵn các phương thức trên Hub)

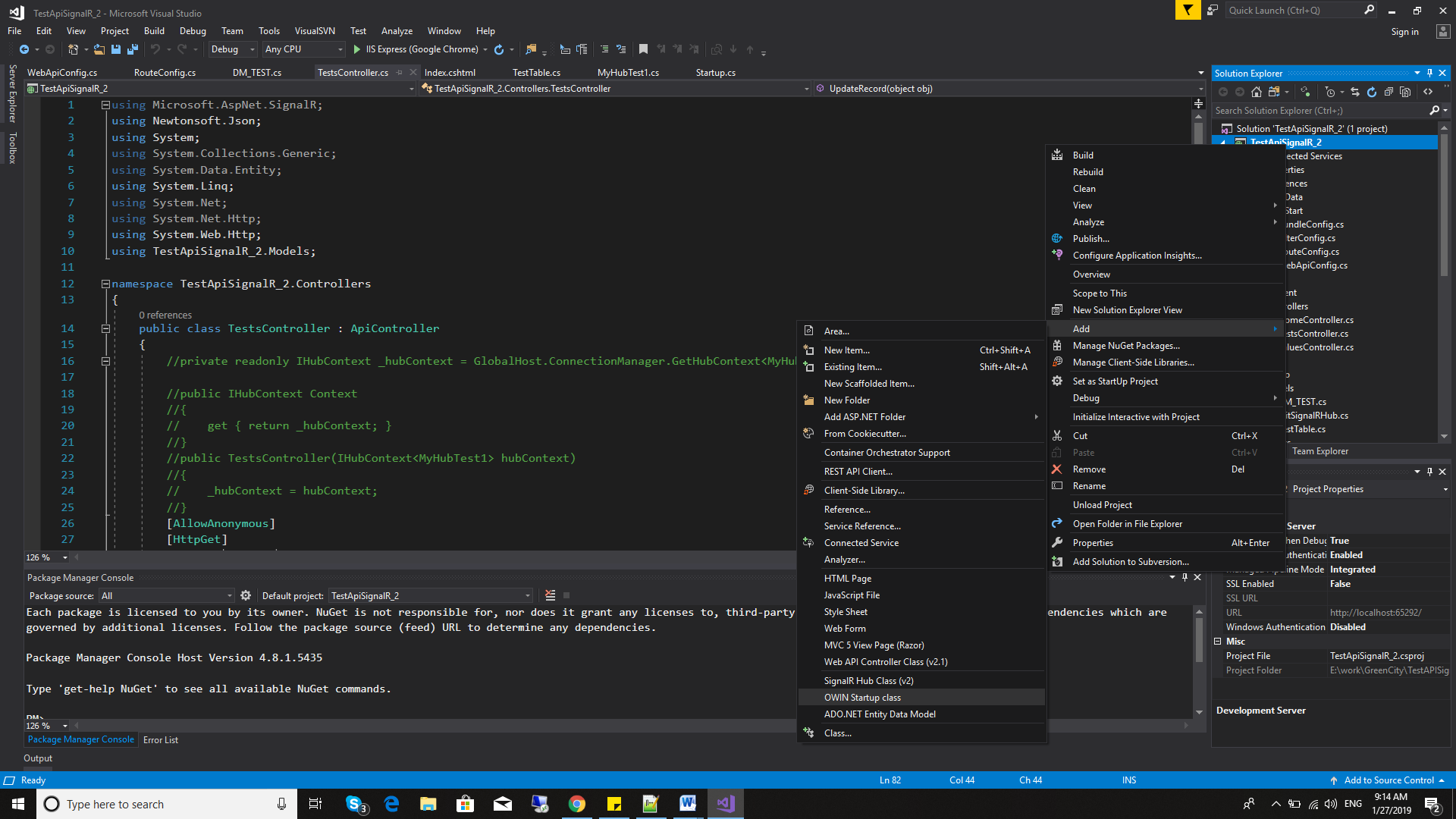
Khi bất kỳ client nào gọi thực thi bất kỳ phương thức nào trong hub thì có thể tất cả các clients được connect với Hub sẽ nhận được thông báo.

1. Hướng dẫn thiết lập

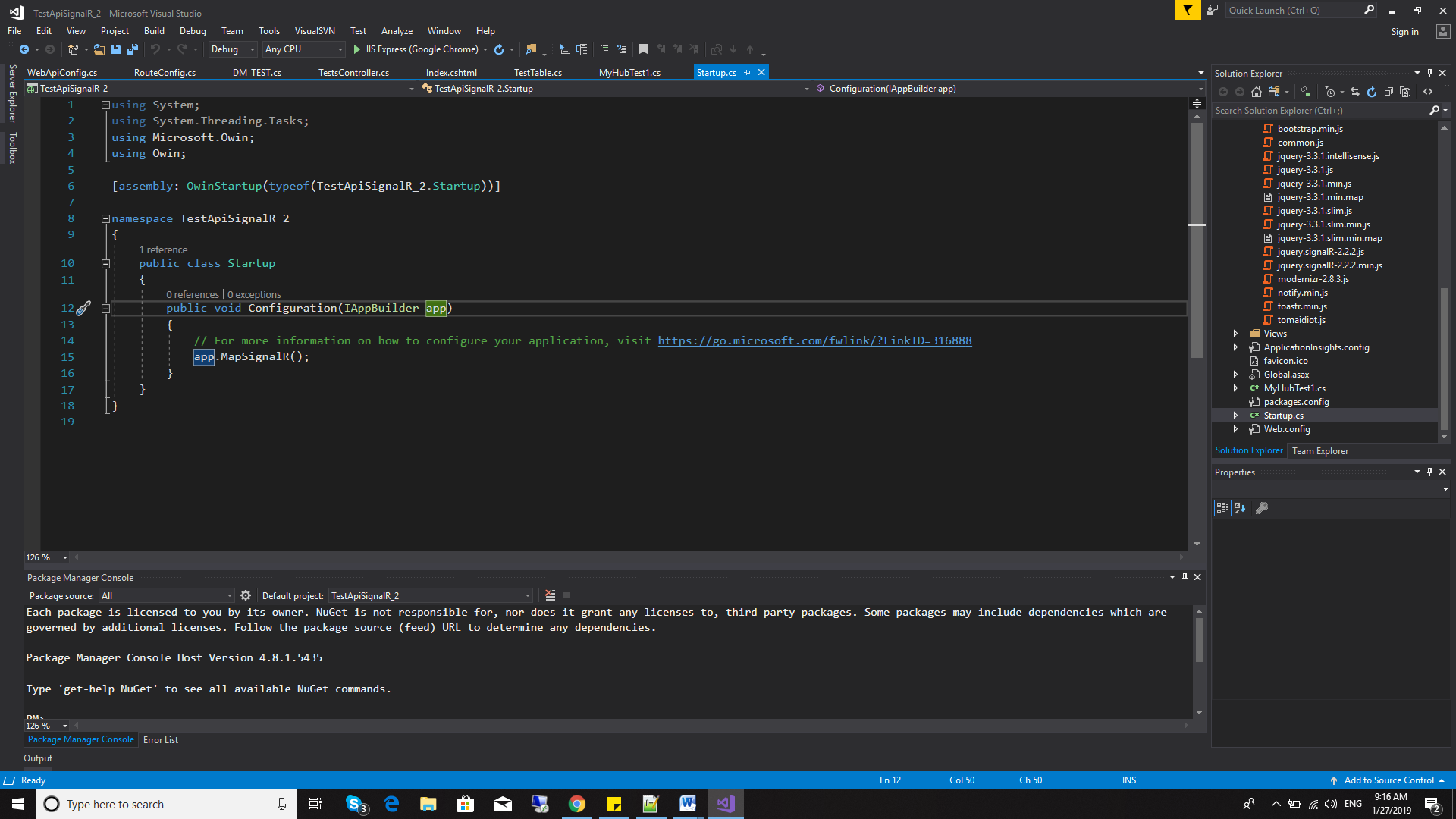
Tạo project Web API MVC

1. Xây dựng Hub
   1. Cài đặt các package

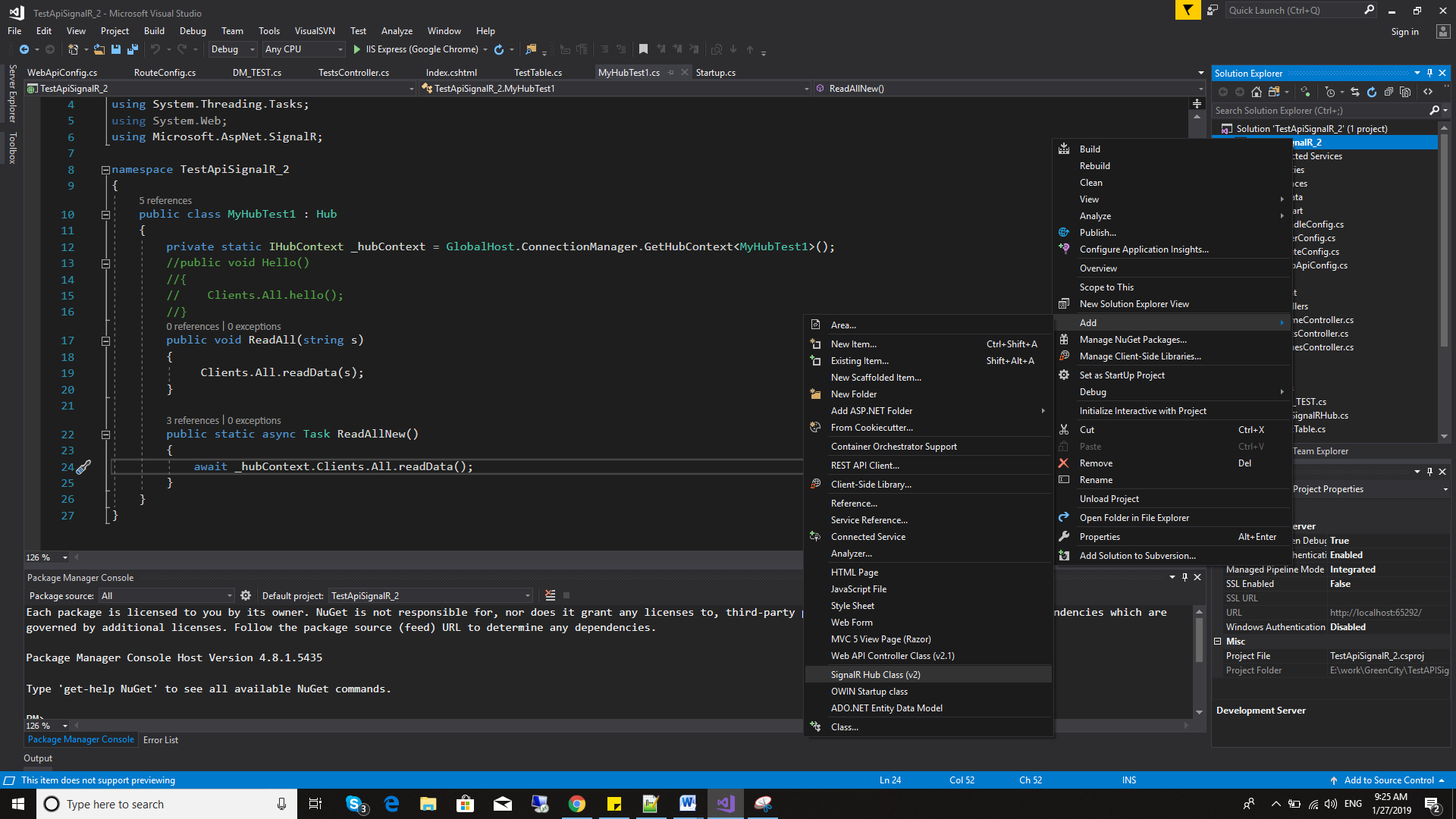
* Microsoft.AspNet.SignalR package
  1. Tạo class Startup.cs kế thừa từ class OWIN

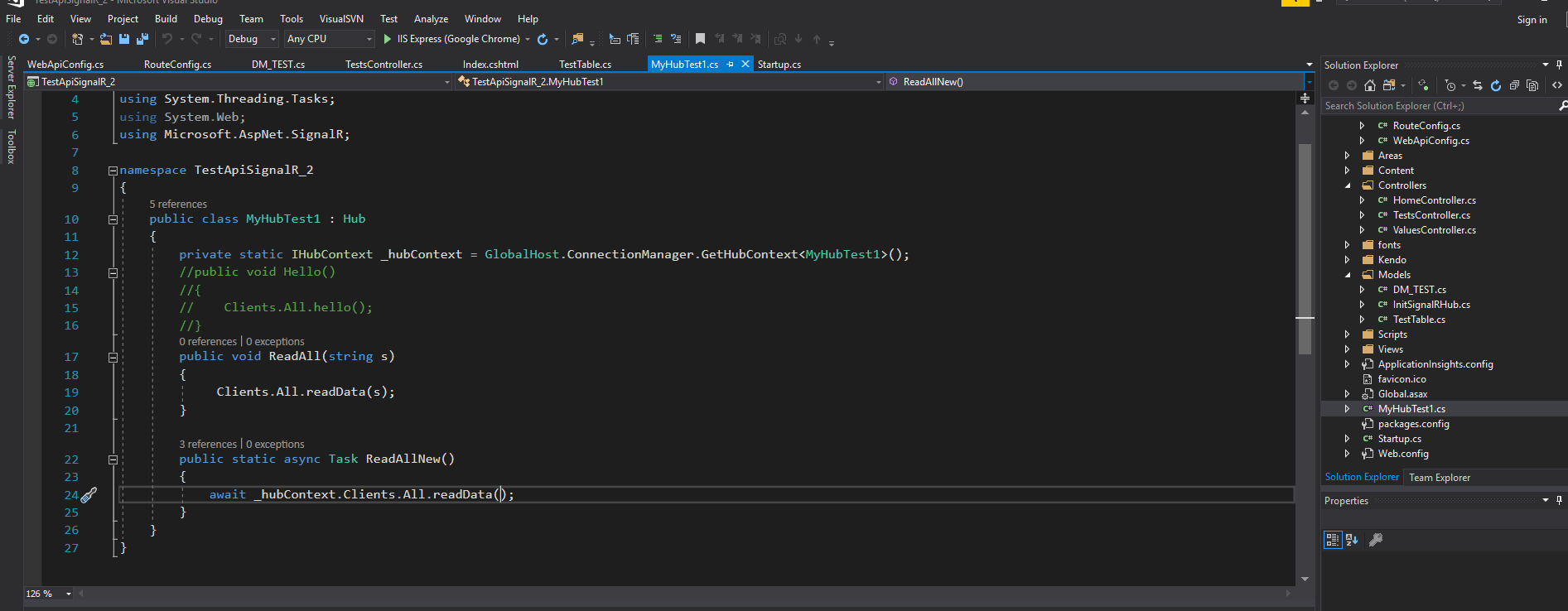


Trong phương thức Configuration của class Startup.cs gọi một phương thức mapping với SignalR



* 1. Tạo class Hub và định nghĩa các phương thức trên server để các clients gọi vào (còn gọi là phương thức client)





Trong hình trên định nghĩa 2 phương thức trên server class Hub: ReadAll và ReadAllNew, trong 2 phương thức này định nghĩa phương thức mà server “trả về” cho các client (được kết nối với Hub signalR), khi được bất kỳ một client nào gọi vào phương thức server trên.

Note: theo sơ đồ trên cần định nghĩa một Interface IhubContext:

private static IHubContext \_hubContext = GlobalHost.ConnectionManager.GetHubContext<MyHubTest1>();

Việc định nghĩa này có chức năng để implement phương thức của Hub vào các phương thức khác trong app -> gửi thông báo đến các client đã kết nối với Hub, khi một client (có thể không kết nối với Hub) thực thi phương thức (mà có implement phương thức của hub vào)

public class MyHubTest1 : Hub

{

private static IHubContext \_hubContext = GlobalHost.ConnectionManager.GetHubContext<MyHubTest1>();

//public void Hello()

//{

// Clients.All.hello();

//}

public void ReadAll(string s) //Phương thức mà client gọi vào server

{

Clients.All.readData(s); //(phương thức server trả lời clients)

}

public static async Task ReadAllNew()//Phương thức mà client gọi vào server

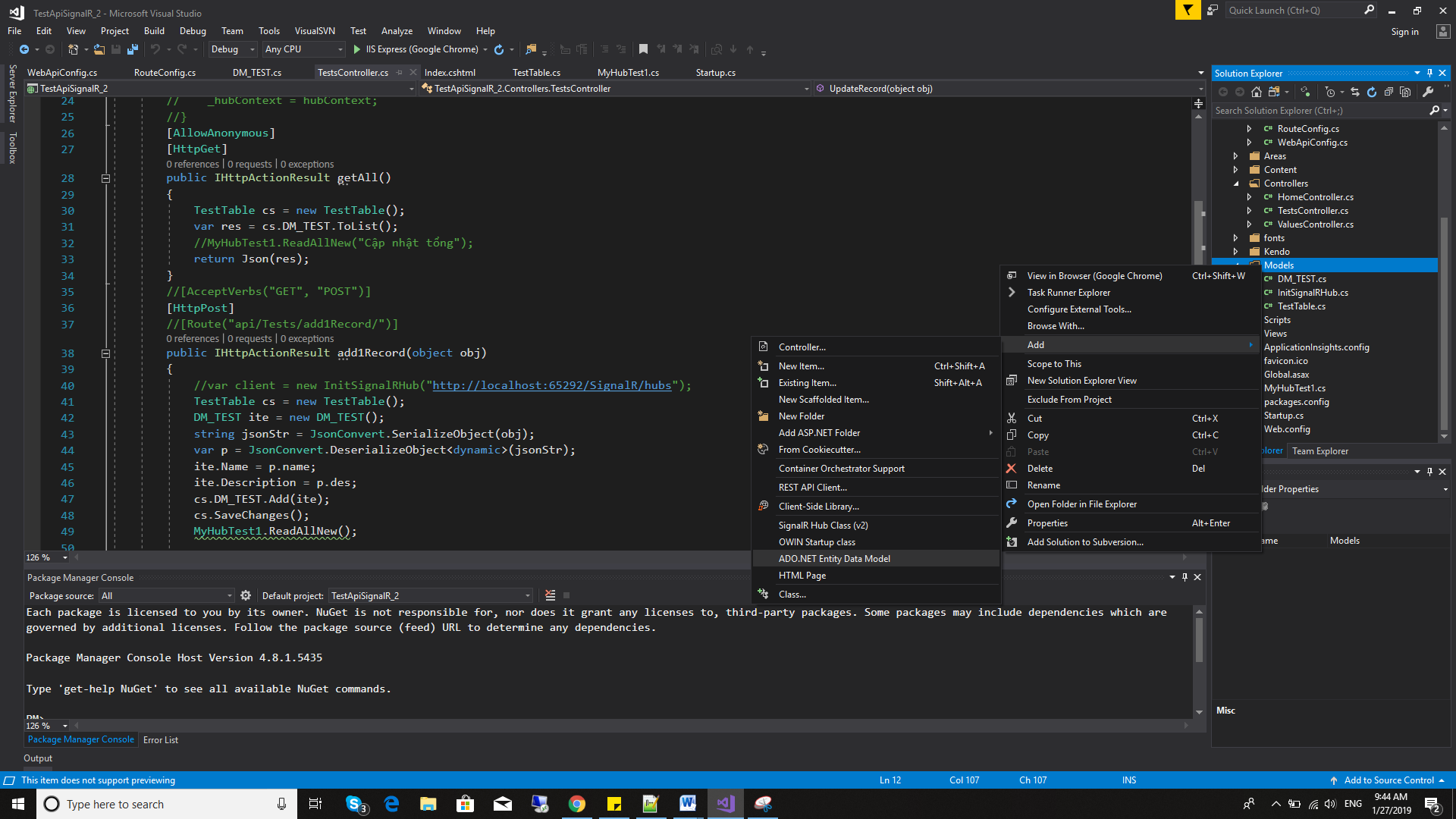
{

await \_hubContext.Clients.All.readData();//(phương thức server trả lời clients)

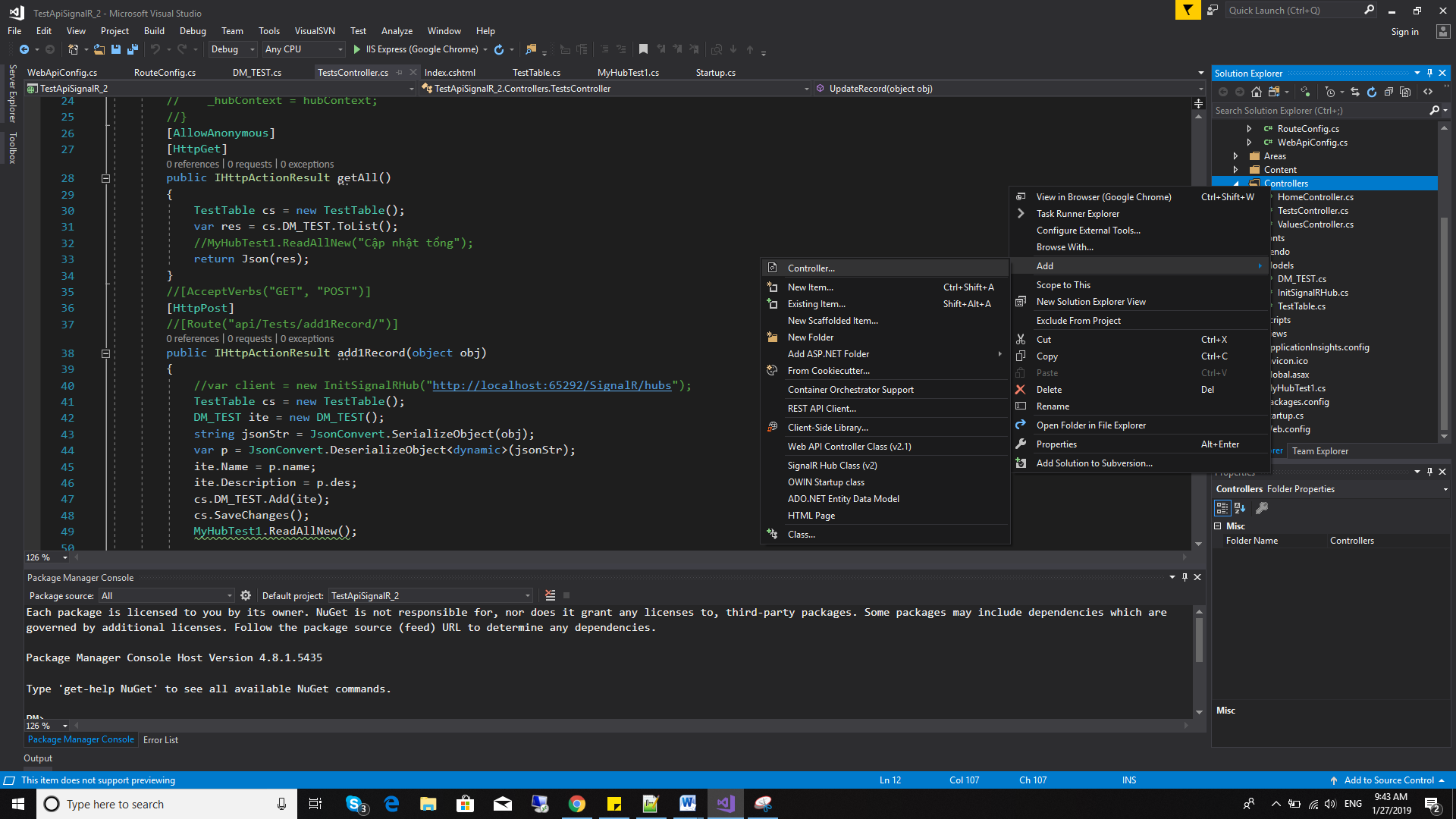
}

}

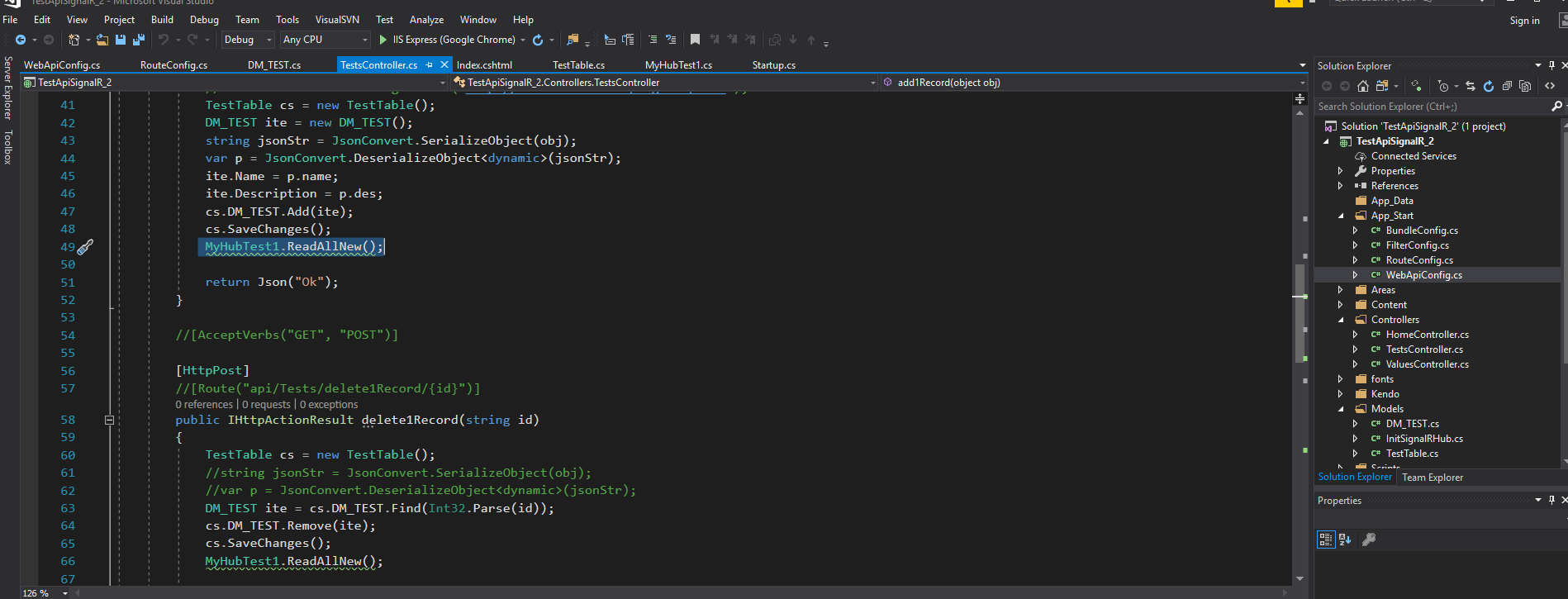
* 1. Tạo các phương thức API
     1. Tạo Model Entity class với cơ sở dữ liệu



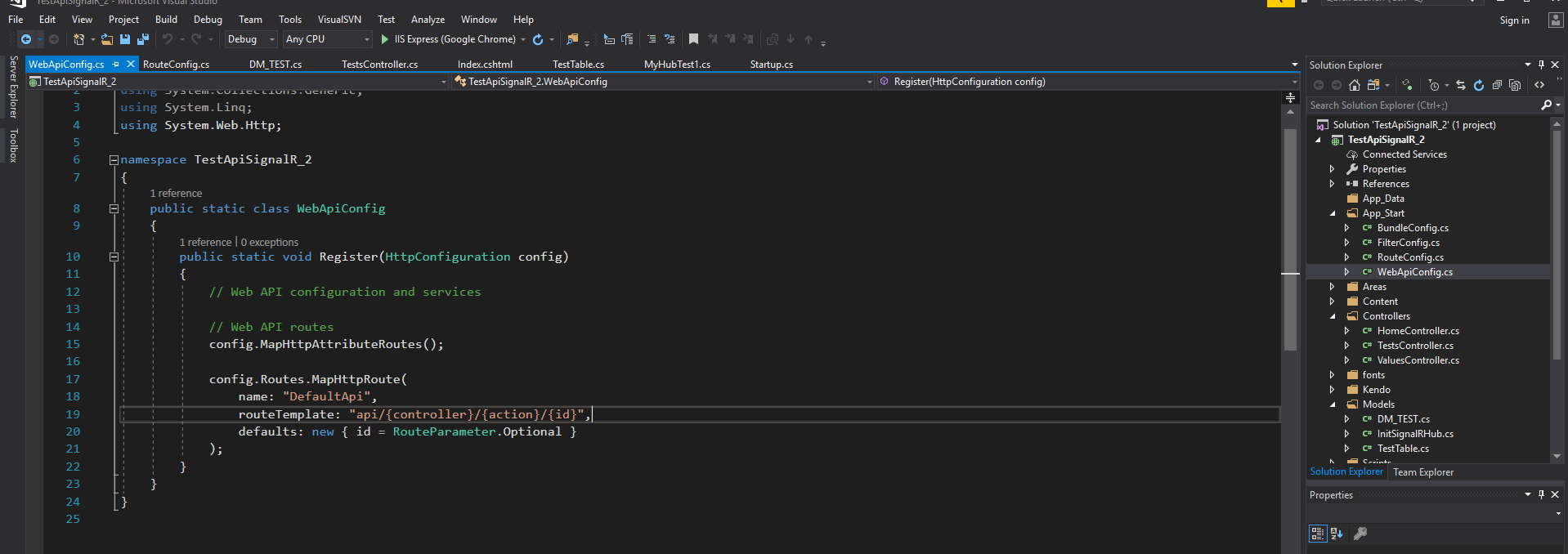
* + 1. Tạo controller và các phương thức



Trong các phương thức của controller API -> implement phương thức server để khi client gọi vào thì sẽ thông báo cho tất cả cho các client được kết nối với HUB.



* + 1. Thay đổi định dạng route webAPI config (quan trọng)\*



“api/{contoller}/{action}/{id}”

* + 1. Thực hiện kết nối client web với SignalR Hub

Trong file js. Thực hiện kết nối với SignalR Hub

C1: Khai báo 1 URL kết nối với Hub:

var hubURL = "http://localhost:65292/SignalR/hubs";

var connection = $.hubConnection(hubURL, { useDefaultPath: false });

var testHubProxy = $.connection.hub.createHubProxy('MyHubTest1');

C2: kết nối với Hub thông qua định nghĩa tên Hub trong class Hub:

var testHubProxy = $.connection.myHub1;

* + 1. Gọi và định nghĩa phương thức của server trả về clients (sau khi clients gọi phương thức server)

testHubProxy.client.readData = function (s) {

//defined method (respone to client from server)

//console.log(s);

tk\_grid.dataSource.read();

console.log("Grid load OK!");

$("#grid").notify("Cập nhật thành công!", { position: "bottom right", className: "success" });

//$('#testInput2').val(s);

}

* + 1. Bắt đầu start kết nối với Hub

$.connection.hub.start().done(function () {

//to do có thể involve các phương thức

//$("#testClick").click(function () {

//call method in server

//testHubProxy.server.readAll("OK");

//alert("OK");

//alert($('#testInput').val());

//});

})

Tài liệu tham khảo:

Web API:

<https://docs.microsoft.com/en-us/aspnet/web-api/overview/getting-started-with-aspnet-web-api/tutorial-your-first-web-api>

<https://docs.microsoft.com/en-us/aspnet/web-api/overview/getting-started-with-aspnet-web-api/action-results>

Sử dụng fidder test web api:

<https://www.c-sharpcorner.com/article/how-to-use-fiddler-with-asp-net-web-api-testing/>

<http://www.deveasy.info/2017/07/su-dung-fiddler-test-rest-api-va-nghe.html>

SignalR Hub:

<https://docs.microsoft.com/en-us/aspnet/signalr/overview/guide-to-the-api/hubs-api-guide-server>

<https://docs.microsoft.com/en-us/aspnet/signalr/overview/getting-started/tutorial-getting-started-with-signalr-and-mvc>