Bắt đầu với SignalR 2

Hướng đẫn này cho bạn biết làm thế nào dung SignalR để tạo một ứng dụng chat thời gian thực. Bạn sẽ thêm SignalR vào một chương trình web ASP.NET và tạo một HTML page để gởi và hiển thị các message

Các phiên bản phần mềm được dùng trong hướng dẫn này

* Visual Studio 2013
* .NET 4.5
* SignalR version 2

# Tổng quan

Hướng dẫn này giới thiệu phát triển SignalR bằng cách xây dựng ứng dụng chat đơn giản trên trình duyệt. Bạn sẽ thêm thư viện SignalR vào một chương trình web APS.NET trống, tạo một hub class cho việc gởi các message đến clients, và tạo một HTML page để các người dùng gởi và nhận chat message.

SignalR là một opensource thư viện .NET dành cho việc xây dựng các chương trình web yêu cầu tương tác trực tiếp và cập nhật dữ liệu thời gian thực. Ví dụ bao gồm các ứng dụng xã hội, game nhiều người chơi, hợp tác kinh doanh, và tin tức, thời tiết, hoặc các ứng dụng cập nhật tài chính. Chúng thường được gọi là các ứng dụng thời gian thực

SignalR đơn giản quá trình xây dựng một ứng dụng thời gian thực. Nó bao gồm một thư viện máy chủ ASP.NET và một thư viện client JavaScript để làm cho nó dễ dàng hơn việc quản lý các kết nối giữa client-server và đẩy các nội dung cập nhật cho các client. Bạn có thr thêm thư viện SignalR vào một ứng dụng ASP.NET hiện có để đặt được tính năng thời gian thực

Hướng dẫn này trình bày các việc cần làm để phát triển chương trình dùng SignalR:

* Thêm thư viện SignalR vào chương trình web ASP.NET
* Tạo một class hub để đấy nội dung về các client
* Tạo mọt class startup OWIN để configure chương trình
* Dùng thư viện jQuery SignalR trong một trang web để gởi các message và hiển thị các cập nhật từ hub

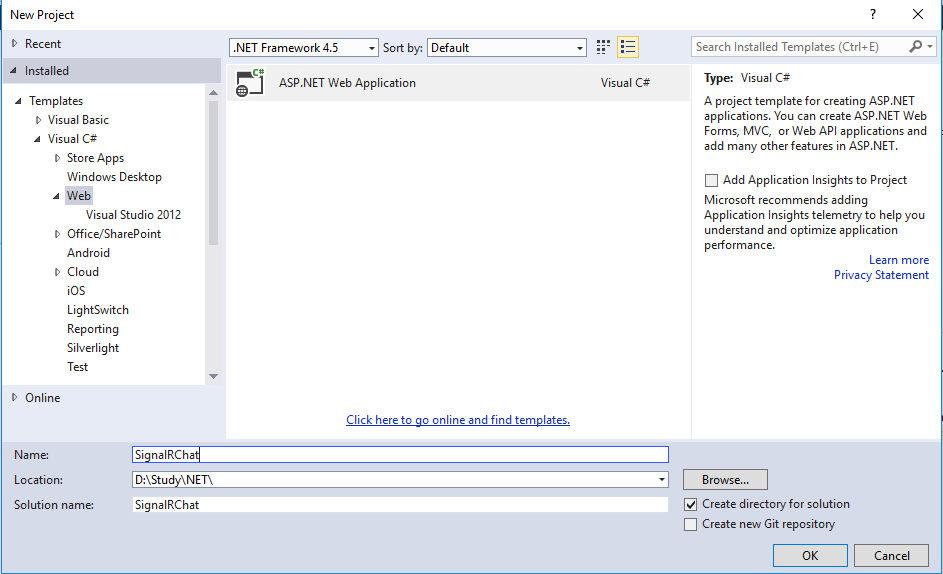
Ảnh chụp màn hình sau đây hiển thị ứng dụng chat đang chạy trên trình duyệt. Mỗi người dùng mới có thể đăng nhận xét và xem nhận xét sau khi người dùng tham gia cuộc trò chuyện.

# Cài đặt Project

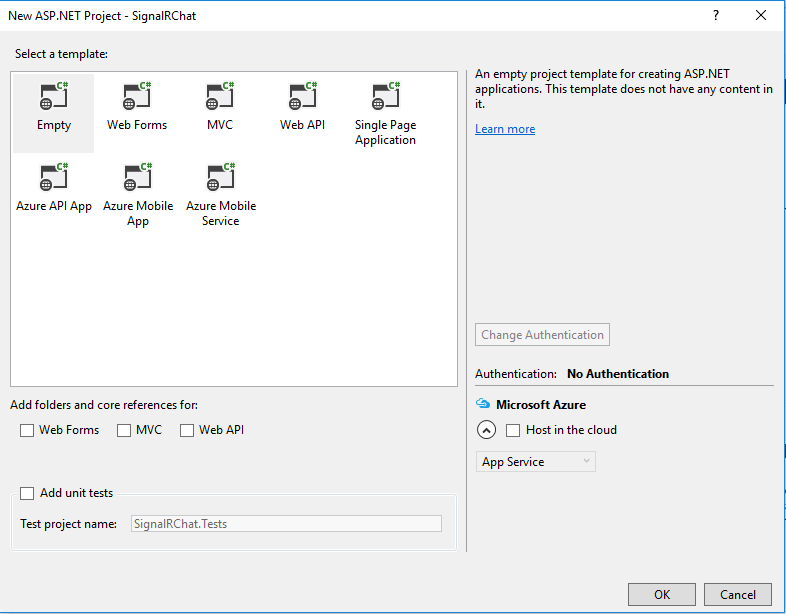
Phần này chỉ ra cách sử dụng Visual 2013 và SignalR 2 để tạo một ứng dụng web ASP.NET rỗng, thêm SignalR và tạo ứng dụng trò chuyện

Các bước sau đây sử dụng Visual Studio 2013 để tạo một ứng dụng web ASP.NET trống và thêm thư viện SignalR:

1. Trong Visual Studio, tạo một ASP.NET Web Application



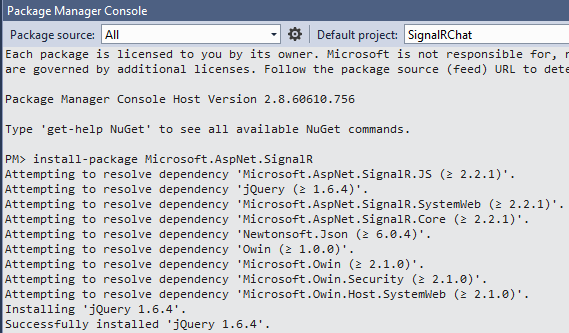
1. Trong cửa sổ New ASP.NET Project, chọn Empty và chọn Create Project



1. Trong Solution Explorer, chuột phải project, chọn Add | SignalR Hub Class (v2). Tên class là ChatHubs.cs và thêm nó vào project. Bước này tạo class ChatHub và thêm các project một tập các file script và các assembly references có hỗ trợ SignalR.

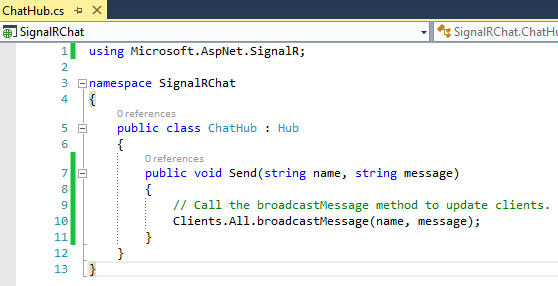
Chạy dòng lệnh sau trong Package Manager Console:

install-package Microsoft.AspNet.SignalR

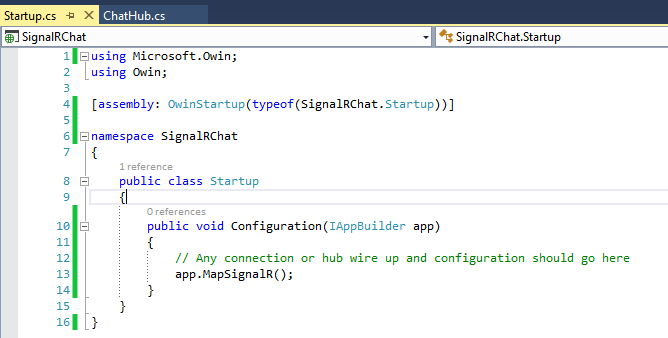


Nếu bạn dùng console thêm SignalR, tạo SignalR hub class như một bước riêng biệt sau khi bạn thêm SignalR (chuột phải project và chọn Add | SignalR Hub Class (v2))

1. Trong Solution Explorer, mở thư mục Scripts, các thư viện Script cho jQuery và SignalR được nhìn thấy trong project.
2. Thay đoạn code trong class mới ChatHub với đoạn code bên dưới.



1. Trong Solution Explorer, chuột phải vào project, sau đó click Add | OWIN Startup Class. Đặt tên class là Startup và click OK
2. Thay đổi các nội dung của Startup class như bên dưới.



1. Trong Solution Explorer, chuột phải project, sau đó click Add | HTML Page. Đặt tên cho trang mới này là index.html
2. Trong Solution Explorer, chuột phải trang HTML bạn vừa tạo và click Set as Start Page
3. Thay code mặc định trong trang này bằng đoạn code sau:

<!DOCTYPE html>

<html>

<head>

<title>SignalR Simple Chat</title>

<style type="text/css">

.container {

background-color: #99CCFF;

border: thick solid #808080;

padding: 20px;

margin: 20px;

}

</style>

</head>

<body>

<div class="container">

<input type="text" id="message" />

<input type="button" id="sendmessage" value="Send" />

<input type="hidden" id="displayname" />

<ul id="discussion">

</ul>

</div>

<!--Script references. -->

<!--Reference the jQuery library. -->

<script src="Scripts/jquery-1.6.4.min.js" ></script>

<!--Reference the SignalR library. -->

<script src="Scripts/jquery.signalR-2.1.0.min.js"></script>

<!--Reference the autogenerated SignalR hub script. -->

<script src="signalr/hubs"></script>

<!--Add script to update the page and send messages.-->

<script type="text/javascript">

$(function () {

// Declare a proxy to reference the hub.

var chat = $.connection.chatHub;

// Create a function that the hub can call to broadcast messages.

chat.client.broadcastMessage = function (name, message) {

// Html encode display name and message.

var encodedName = $('<div />').text(name).html();

var encodedMsg = $('<div />').text(message).html();

// Add the message to the page.

$('#discussion').append('<li><strong>' + encodedName

+ '</strong>:&nbsp;&nbsp;' + encodedMsg + '</li>');

};

// Get the user name and store it to prepend to messages.

$('#displayname').val(prompt('Enter your name:', ''));

// Set initial focus to message input box.

$('#message').focus();

// Start the connection.

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

$('#sendmessage').click(function () {

// Call the Send method on the hub.

chat.server.send($('#displayname').val(), $('#message').val());

// Clear text box and reset focus for next comment.

$('#message').val('').focus();

});

});

});

</script>

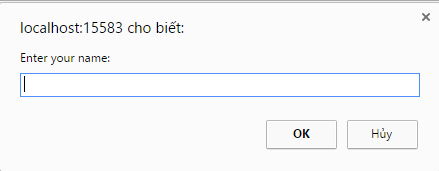
</body>

</html>

1. Save All project

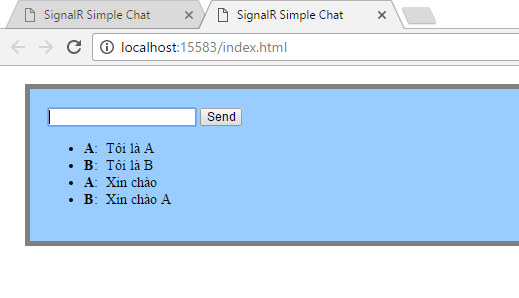
# Chạy ví dụ

1. Bấm F5 để chạy project ở chế độ debug. Trang HTML tải lên trình duyệt và hộp thoại nhập tên người dùng xuất hiện



1. Nhập 1 tên
2. Copy URL của trình duyệt và dùng nó để mở nhiều tab khác. Trong mỗi tab, nhập một tên duy nhất
3. Trong mỗi trình duyệt, thêm một comment và click Send. Comment sẽ hiển thị trong tất cả các trình duyệt còn lại.

Ảnh chụp màn hình sau đây hiển thị ứng dụng chạy trên 3 tab trình duyệt khác nhau, tất cả đều được cập nhật khi một cá nhân gởi 1 thông báo



Giải thích Code

Ứng dụng chat Signal thẻ hiện hai nhiệm vụ phát triển Signal cơ bản: Tạo ra một hub như là đối tượng điều phối chính trên máy chủ, và dùng thư viện jQuery SignalR để gởi và nhận các message

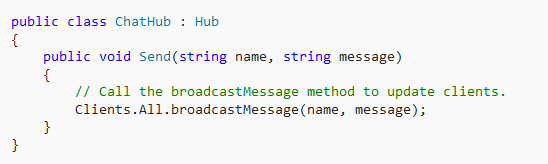
Các SignalR Hubs

Trong code ví dụ, ChatHub class kế thừa từ class Microsoft.AspNet.SignalR.Hub. Kế thừa từ class Hub là một cách hữu ích để xây dựng một ứng dụng SignalR. Bạn có thể tạo các phương thức public trên class hub và sau đó truy cập phương thức đó bằng cách gọi chúng từ script trong một trang web

Trong chat code, các client gọi phương thức ChatHub.Send để gởi một message mới. Hub lần lượt gởi message đến tất cả các client bằng cách gọi Client.All.broadcastMessage.

Phương thức Send phải có những phần sau:

* Dùng phương thức public trên một hub để các client có thể gọi chúng
* Dùng Microsoft.AspNet.SignalR.Hub.Clients để truy cập tất cả client được connect tới hub
* Gọi một hàm trên client (như broadcastMessage) để cập nhật cho các client



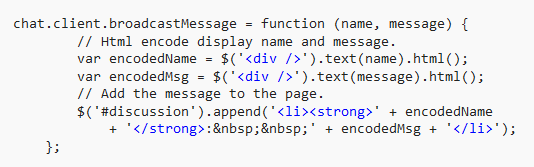
SignalR và jQuery

Trang HTML trong code ví dụ cho biết làm thể nào để dung thư viện jQuery SignalR để truyền thông với một SignalR hub. Các nhiệm vụ trong mã diễn tả một proxy tham chiếu đến hub, diễn tả một hàm server có thể gọi để đẩy nội dung đến các client, và bắt đầu kết nối để gởi các message tới hub.

Code sau mô tả một tham chiếu đến một hub proxy



Đoạn mã dưới đây chỉ ra cho bạn làm thế tạo để tạo một lời gọi hàm trong script. Hub class trên server gọi hàm này để đẩy nội dung update đến mỗi client. 2 dong HTML encode nội dung trước khi hiển thị nó là tùy chọn và hiển thị một cách đơn giản để ngăn chặn script injection.



Đoạn mã dưới đây chỉ ra cách mở một kết nối với hub. Code bắt đầu kết nối và chuyển nó một hàm để xử lý sự kiện click trên button Send trên trang web

