

Thứ 2 đi GrabCar miễn phí

Nhập mã WOW khi đặt GrabCar, miễn phí 2 chuyến (tối đa 50k/ chuyến). Click để xem thêm!



ASP.NET SignalR

Suraj Kumar Maddheshiya 05:54:00 MVC 2 comments

ASP.NET SignalR is a library for ASP.NET developers that simplifies the process of adding real-time web functionality to applications. Real-time web functionality is the ability to have server code push content to connected clients instantly as it becomes available, rather than having the server wait for a client to request new data.

SignalR can be used to add any sort of "real-time" web functionality to your ASP.NET application. While chat is often used as an example, you can do a whole lot more. Examples include dashboards and monitoring applications, collaborative applications (such as simultaneous editing of documents), job progress updates, and real-time forms.

Now a days, Most of the web applications are multiuser application, where multiple users connected at the same time. But the problem is when any user does any changes (ex. insert a new record) other users doesn't know it unless the request is initiated by the user for seeking the any updates.

So, we must have a way to notify all the connected clients if there is any changes happens on the server without a refresh or update the web page. This is the scenario in which asp.net SignalR comes.

Creating a push notification system with SignalR

Follow the below steps in order to implement "push notification system with SignalR in asp.net MVC".

Here In this Example, I am using Visual Studio 2012

Step-1: Create New Empty MVC Project.

Step-2: Add a SQL Server Database.

```
Create database SignalRDB;
```

```
Create table tblEmployee (
```

DOTNET GURU ~

```
)
```

Step-3: Enable Service Broker on the database.

```
ALTER DATABASE SignalRDB SET ENABLE_BROKER WITH ROLLBACK IMMEDIATE ;
```

Step-4: Add Entity Data Model.

Go to Solution Explorer (Visual studio) > Right Click on Project name form Solution Explorer > Add > New item > Select ADO.net Entity Data Model under data > Enter model name > Add.

Search

Like Share 139 people like this. Sign Up to see what your

Air China Limited Offer

airchina.vn

Special Offer Only On Air China Official Site. Time Limited, Book Now!



Bạn muốn sống thoải mái? - Bắt đầu giao dịch với FBS

vnfbs.com

Thành công và sự giàu có chỉ cách bạn một bước!



TOPICS

ADO .Net	Ajax	Angular Js
ASP .Net	C#	CSS DI
Dotnet	Entity Framework	
Html	Javascript	Jquery
Lambda	Linq	MVC
Sql Server	SSRS	WCF
Web Api	Web Service	XML

PAGES

[Home](#) [.Net](#) [MVC](#) [WCF](#) [Sql Server](#) [AngularJS](#)
[LINQ Operators and Lambda Expression](#)
[LINQ & Lambda Expressions](#)
[Important Links](#)
[Using a GUID as an EntityKey in Entity Framework 4](#)
[MVC 3 Modeling with Models \(Validation using Data Annotations\)](#)
[Creating a basic ASP.NET MVC Application](#)

A popup window will come (Entity Data Model Wizard) > Select Generate from database > Next >

Chose your data connection > select your database > next > Select tables > enter Model Namespace > Finish.

Step-5: Install SignalR NuGet Package.

Solution Explorer > Right Click on References > Manage NuGetPackages > Search for "SignalR"> Install > Close.

Or you can also install from package manager console.

Go to Tools (top menu) > Library Package Manager > Open "Package Manager Console" Type below command

```
PM> Install-Package Microsoft.AspNet.SignalR
```

Step-6: Add an Owin startup file.

Add an Owin startup class in your application for enabling the SignalR in our application. Add a new class in the project named "Startup.cs"

Write following code in Startup.cs file.

```
using Microsoft.Owin;
using Owin;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
[assembly: OwinStartup(typeof(SignalRDemo.Startup))]
namespace SignalRDemo
{
    public class Startup
    {
        public void Configuration(IAppBuilder app)
        {
            app.MapSignalR();
        }
    }
}
```

Step-7: Add a SignalR hub class.

Now, you need to create a SignalR Hub class, this makes possible to invoke the client side JavaScript method from the server side. In this application, we will use this for showing

DOTNET GURU ~

[Home](#)
[.Net](#)
[MVC](#)
[WCF](#)
[Sql Server](#)
[AngularJS](#)

Add a new class named "NotificationHub.cs"

Enter the following contents into the file.

```
using Microsoft.AspNet.SignalR;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
namespace SignalRDemo
{
    public class NotificationHub : Hub
    {
        //Nothing required here
    }
}
```

[jQuery getJSON Example in ASP.NET MVC](#)

[Using JsonResult in ASP.NET MVC](#)

[jQuery Selectors & Methods](#)

[How to send email in ASP.NET C# MVC 3](#)

[Dependency Injection in C# .NET](#)

[SQL Server 2005/2008 Functions](#)

[ASP .Net MVC Interview Questions](#)

[Ado .Net InterView Questions](#)

[ASP .Net Interview Question](#)

[InterView Programming](#)

[.Net Interview Questions](#)

[Collections](#)

[WCF Interview Questions](#)

[SQLServer Interview Questions](#)

[SQL - Query Related Interview Questions](#)

[Angularjs Interview Questions and Answers](#)

[AngularJS Interview Questions & Answers](#)

[javascript-interview-questions](#)

[ADO.NET Entity Framework Interview Questions](#)

DOTNET GURU ARCHIVES

▼ 2016 (11)

► Aug 2016 (2)

▼ Jul 2016 (1)

[ASP.NET SignalR](#)

► Jun 2016 (1)

► May 2016 (1)

► Mar 2016 (2)

► 2015 (42)

► 2014 (28)

► 2013 (50)

```
//public void Hello()
//{
//  Clients.All.hello();
//}
}
```

you can see above, the NotificationHub.cs class is empty. Left the class empty as we will use the class later from another place.

Step-8: Add connection string into the web.config file

Open the application root Web.config file and find the element. Add the following connection string to the element in the Web.config file.

```
<add name="sqlConString" connectionString="data source=VL32-PC-SURAJ\SQLEXPRESS;
```

Step-9: Add another class file for register notification for data changes in the database

In this class, you need to create a SQL dependency which allows your application to be notified when a data has changed in the database (Microsoft SQL Server). Write the following in this class...

1. RegisterNotification- void method for register notification
2. SqlDependency_OnChange- SqlDependency onchnage event, get fired when assigned SQL command produced a different
3. GetContacts- This is a method for return the changes happened on the server, here our new inserted contact data

```
using Microsoft.AspNet.SignalR;
using System;
using System.Collections.Generic;
using System.Configuration;
using System.Data.SqlClient;
using System.Linq;
using System.Web;
namespace SignalRDemo
{
    public class NotificationComponent
    {
        //Here we will add a function for register notification (will add sql depende
        public void RegisterNotification(DateTime currentTime)
        {
            string conStr = ConfigurationManager.ConnectionStrings["sqlConString"].Conn
            string sqlCommand = @"SELECT [ID],[Name] from [dbo].[tblEmployee] where [Ac
            //you can notice here I have added table name like this [dbo].[Contacts] wi
            using (SqlConnection con = new SqlConnection(conStr))
```

DOTNET GURU ~

[Home](#)
[.Net](#)
[MVC](#)
[WCF](#)
[Sql Server](#)
[AngularJS](#)

```
SqlConnection con = new SqlConnection(conStr), cmd = new SqlCommand(),
if (con.State != System.Data.ConnectionState.Open)
{
    con.Open();
}
cmd.Notification = null;
SqlDependency sqlDep = new SqlDependency(cmd);
sqlDep.OnChange += sqlDep_OnChange;
//we must have to execute the command here
using (SqlDataReader reader = cmd.ExecuteReader())
{
    // nothing need to add here now
}
}
}
void sqlDep_OnChange(object sender, SqlNotificationEventArgs e)
```

```

{
    //or you can also check => if (e.Info == SqlNotificationInfo.Insert) , if y
    if (e.Type == SqlNotificationType.Change)
    {
        SqlDependency sqlDep = sender as SqlDependency;
        sqlDep.OnChange -= sqlDep_OnChange;
        //from here we will send notification message to client
        var notificationHub = GlobalHost.ConnectionManager.GetHubContext<Notifica
        notificationHub.Clients.All.notify("added");
        //re-register notification
        RegisterNotification(DateTime.Now);
    }
}
public List<tblEmployee> GetData(DateTime afterDate)
{
    using (SignalRDBEntities dc = new SignalRDBEntities())
    {
        return dc.tblEmployees.Where(a => a.AddedOn > afterDate).OrderByDescendin
    }
}
}
}

```

Step-10: Create an MVC Controller

Add a new Controller named "HomeController.cs"

Step-11: Add new action into your controller.

Here I have added "Index" Action into "Home" Controller.

```

public ActionResult Index()
{
    return View();
}

```

Step-12: Add view for your Action named "Index.cshtml"

```

@{
    ViewBag.Title = "Index";
}

<h2>Index</h2>

<a target="_blank" href="Add/">Add Employee</a>

```

DOTNET GURU ~

[Home](#)
[.Net](#)
[MVC](#)
[WCF](#)
[Sql Server](#)
[AngularJS](#)

```

public JsonResult GetNotifications()
{
    var notificationRegisterTime = Session["LastUpdated"] != null ? Convert.ToDate
    NotificationComponent NC = new NotificationComponent();
    var list = NC.GetData(notificationRegisterTime);
    //update session here for get only new added contacts (notification)
    Session["LastUpdate"] = DateTime.Now;
    return new JsonResult { Data = list, JsonRequestBehavior = JsonRequestBehavior
}
}

```

Step-14: Add and update _Layout.cshtml for showing notification.

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>@ViewBag.Title - MY SignalR Demo</title>
  <link href="~/Content/Site.css" rel="stylesheet" type="text/css" />
  <link href="~/Content/bootstrap.min.css" rel="stylesheet" />
  <script src="~/Scripts/modernizr-2.6.2.js"></script>
</head>
<body>
  <div class="navbar navbar-inverse navbar-fixed-top">
    <div class="container">
      <div class="navbar-header">
        <span class="noti glyphicon glyphicon-bell"><span class="count">&nbsp;</span></span></div>
        <div class="noti-content">
          <div class="noti-top-arrow"></div>
          <ul id="notiContent"></ul>
        </div>
        <button type="button" class="navbar-toggle" data-toggle="collapse" data-target="#navbar-collapse">
          <span class="icon-bar"></span>
          <span class="icon-bar"></span>
          <span class="icon-bar"></span>
        </button>
        @Html.ActionLink("SignalR Demo", "Index", "Home", null, new { @class = "navbar-brand" })
      </div>
      <div class="navbar-collapse collapse">
        <ul class="nav navbar-nav">
          <li><a href="#">Home</a></li>
          <li><a href="#">Index</a></li>
          <li><a href="#">About</a></li>
          <li><a href="#">Contact</a></li>
        </ul>
      </div>
    </div>
  </div>
  <div class="container body-content">
    @RenderBody()
    <hr />
    <footer>
      <p>&copy; @DateTime.Now.Year - My SignalR Demo</p>
    </footer>
  </div>
  @* Add JQuery Library *@
  <script src="~/Scripts/jquery-2.2.3.min.js"></script>
  <script src="~/Scripts/jquery.signalR-2.2.0.min.js"></script>
  <script src="/signalr/hubs"></script>
  <script src="~/Scripts/bootstrap.min.js"></script>
  @* Add css *@
  <link href="~/Content/bootstrap.min.css" rel="stylesheet" />
<style type="text/css">
  /*Added css for design notification area, you can design by your self*/
  /* COPY css content from youtube video description*/
  .noti-content{
    position:fixed;
    right:100px;
    background:#e5e5e5;
    border-radius:4px;
    top:47px;
  }
  ul#notiContent li {
    margin:3px;
    padding:6px;
    background:#fff;
  }
  .noti-top-arrow{
    border-color:transparent;
  }

```

DOTNET GURU ~

[Home](#)
[.Net](#)
[MVC](#)
[WCF](#)
[Sql Server](#)
[AngularJS](#)

```

}
ul#notiContent{
  max-height:200px;
  overflow:auto;
  padding:0px;
  margin:0px;
  padding-left:20px;
}
ul#notiContent li {
  margin:3px;
  padding:6px;
  background:#fff;
}
.noti-top-arrow{
  border-color:transparent;
}

```

```

border-bottom-color:#F5DEB3;
border-style:dashed dashed solid;
border-width: 0 8.5px 8.5px;
position:absolute;
right:32px;
top:-8px;
}
span.noti{
color:#FF2323;
margin:15px;
position:fixed;
right:100px;
font-size:18px;
cursor:pointer;
}
span.count{
position:relative;
top:-3px;
}
</style>
@* Add jquery code for Get Notification & setup signalr *@
<script type="text/javascript">
$(function () {
// Click on notification icon for show notification
$('#span.noti').click(function (e) {
debugger;
e.stopPropagation();
$('#.noti-content').show();
var count = 0;
count = parseInt($('#span.count').html()) || 0;
//only load notification if not already loaded
if (count > 0) {
updateNotification();
}
$('#span.count', this).html('&nbsp;');
})
// hide notifications
$('#html').click(function () {
$('#.noti-content').hide();
})
// update notification
function updateNotification() {
$('##notiContent').empty();
$('##notiContent').append($('

```

DOTNET GURU ~

[Home](#)
[.Net](#)
[MVC](#)
[WCF](#)
[Sql Server](#)
[AngularJS](#)

```

        console.log(error);
    }
})
}
// update notification count
function updateNotificationCount() {
var count = 0;
count = parseInt($('#span.count').html()) || 0;
count++;
$('#span.count').html(count);
}
// signalr js code for start hub and send receive notification
var notificationHub = $.connection.notificationHub;
$.connection.hub.start().done(function () {
console.log('Notification hub started');

```

```

    });
    //signalr method for push server message to client
    notificationHub.client.notify = function (message) {
        if (message && message.toLowerCase() == "added") {
            updateNotificationCount();
        }
    }
    })
</script>
</body>
</html>

```

Step-15: Update global.asax.cs to start, stop SQL dependency

```

using System;
using System.Collections.Generic;
using System.Configuration;
using System.Data.SqlClient;
using System.Linq;
using System.Web;
using System.Web.Http;
using System.Web.Mvc;
using System.Web.Routing;
namespace SignalRDemo
{
    // Note: For instructions on enabling IIS6 or IIS7 classic mode,
    // visit http://go.microsoft.com/?LinkId=9394801
    public class MvcApplication : System.Web.HttpApplication
    {
        string con = ConfigurationManager.ConnectionStrings["sqlConString"].ConnectionString;
        protected void Application_Start()
        {
            AreaRegistration.RegisterAllAreas();
            // WebApiConfig.Register(GlobalConfiguration.Configuration);
            // FilterConfig.RegisterGlobalFilters(GlobalFilters.Filters);
            RouteConfig.RegisterRoutes(RouteTable.Routes);
            //here in Application Start we will start Sql Dependency
            SqlDependency.Start(con);
        }
        protected void Session_Start(object sender, EventArgs e)
        {
            NotificationComponent NC = new NotificationComponent();
            var currentTime = DateTime.Now;
            HttpContext.Current.Session["LastUpdated"] = currentTime;
            NC.RegisterNotification(currentTime);
        }
        protected void Application_End()
        {
            //here we will stop Sql Dependency
            SqlDependency.Stop(con);
        }
    }
}

```

DOTNET GURU ~

[Home](#)

[.Net](#)

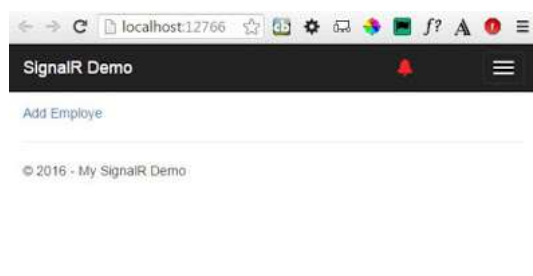
[MVC](#)

[WCF](#)

[Sql Server](#)

[AngularJS](#)

Step-16: Run Application.



Step-17: Now add a new Controller/Action To add Employee Data.

1- I have created AddController.cs in Controller directory.

2- Add action named "Index"

```
public ActionResult Index()
{
    return View();
}
```

3- Create View for Index action to add Employee

```
@model SignalRDemo.tblEmployee
@{
    ViewBag.Title = "Index";
}
<h2>Add Employee</h2>
@using (Html.BeginForm()) {
    @Html.ValidationSummary(true)
    <fieldset>
        <legend>Add Employee</legend>
        <div class="editor-label">
            @Html.LabelFor(model => model.Name)
        </div>
        <div class="editor-field">
            @Html.EditorFor(model => model.Name)
            @Html.ValidationMessageFor(model => model.Name)
        </div>
        <p>
            <input type="submit" value="Create" />
        </p>
    </fieldset>
}
```

4- Add Post action for Index and add code to insert Employee

```
[HttpPost]
public ActionResult Index(tblEmployee model)
{
    SignalRDBEntities entity = new SignalRDBEntities();
    model.AddedOn = DateTime.Now;
    entity.tblEmployees.Add(model);
    entity.SaveChanges();
    return View();
}
```

DOTNET GURU ~

[Home](#)

[.Net](#)

[MVC](#)

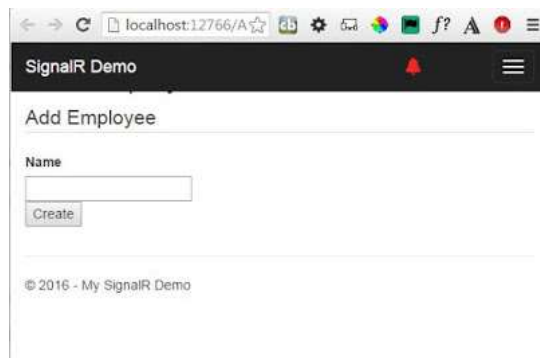
[WCF](#)

[Sql Server](#)

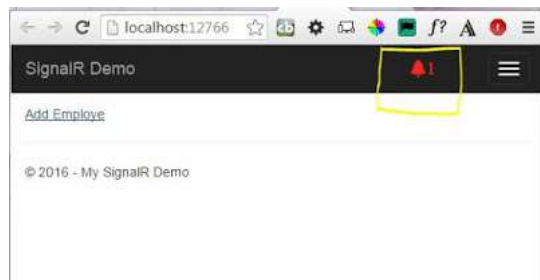
[AngularJS](#)

Step-18: Run Application.

Click On Add Employee Link. it will open a new Page.



Type Name in textbox and click on Create button, it saves data in the table. Now go to first tab to see notification.



Click here to download this project [Download](#)

Like and share if you liked this post.

Thanks-

Share This: [Facebook](#) [Twitter](#) [Google+](#) [Stumble](#) [Digg](#)

[G+](#) Recommend this on Google



[Suraj Kumar Maddheshiya](#)

Myself Suraj Experiencing with Microsoft Technologies.

Related Posts:



[ViewData VS ViewBag Vs TempData in MVC](#)

Introduction In Asp.Net MVC there are three ways to pass/store data between the controllers and views. ViewData ViewData is used to pass data from co... [Read More](#)

DOTNET GURU ~

[Home](#)

[.Net](#)

[MVC](#)

[WCF](#)

[Sql Server](#)

[AngularJS](#)



[~Difference between Lazy Loading and Eager Loading~ In LINQ and Entity](#)

Framework, you have Lazy Loading and Eager Loading for loading the related en... [Read More](#)



[Custom Filters in MVC](#)

~ Custom Filters in MVC - Authorization , Action, Result, Exception Filters ~ Filters in MVC are attributes which you can apply to a controller acti... [Read More](#)



[Dependency Injection](#)

We know that Dependency Injection is software design pattern that allow us to implement loosely couple architecture. The term "loosely couple" carri... [Read More](#)



MVC3 Vs MVC4 Vs MVC5 Vs MVC6

Microsoft has added exciting features in every new version of ASP.NET MVC that make developers more comfortable building scalable web applications ea... [Read More](#)

[Newer Post](#)[Home](#)[Older Post](#)

2 comments:



Dilip Kumar Singh 5 July 2016 at 11:44

Nice one. Keep it up

[Reply](#)

[Replies](#)



Suraj Kumar Maddheshiya 8 July 2016 at 04:29

THankyou Dilip

[Reply](#)

Comment as: Tuan Pham (Gi ▾)

Sign out

Publish

Preview

☐ Notify me

Subscribe to: [Post Comments \(Atom\)](#)

Tạo Email Chuyên Nghiệp

→

?

Dễ Đăng Tạo Tài Khoản Email Riêng. Xử Lý Email Từ Mọi Nơi



DOTNET GURU ~

[Home](#)[.Net](#)[MVC](#)[WCF](#)[Sql Server](#)[AngularJS](#)