CODE DIGEST THE SOLUTION EXPLORER (/)

Featured:

Integrate Google (Gmail) Authentication in Asp.Net MVC 5.0 - OAuth 2.0

05-04-2017

Tagged as: **Asp.Net MVC (/tagged/asp.net%20mvc)** **Posted By** Bala Murugan (/user/2/bala-murugan)

Like Share Be the first of your friends to like this.

In an era of Social Media, the web has developed so much and so the authentication systems to access services on the web. People nowadays are more comfortable using their social media identities or any other identities from providers like Gmail, Microsoft, etc. to login to various other services across the web. This is generally called as Single Sign-on. Using these types of identity providers to login to an external services has benefits to users as wells the service providers. Users need not maintain or remember separate username/passwords of different services. Nowadays, almost everyone has membership in either social media platforms or Gmail or Microsoft or any other Identity providers (OpenID, etc.). On the other hand, when a service provider is allowing users to use a third party login then it relieves them securing user details and they are tend to get more customers to use their service. Moving forward, let us see how to integrate GMAIL authentication using OAuth 2.0 in our Asp.Net MVC 5.0 application. Read here (https://oauth.net/2/) to know more about OAuth.

Note *– This article uses Asp.Net MVC 5.0 without OWIN and Asp.Net Identity. We will see how to integrate these external identity providers in another article.*

OAuth Integration Components

All the providers including Google maintains a developer site to help us integrate their OAuth service into our application. In this article, let us use a commonly used component called DotnetOpenAuth to integrate Google authentication into our Asp.Net MVC application. Visual Studio project templates by default include the DotNetOpenAuth Nuget packages and a default AccountController implementation to start using the common OAuth providers like Google, FaceBook, Microsoft and Twitter. You need to select "Internet Application" or "Individual User Accounts" as project type when creating the project to get this default implementation. So, using the starter project will give you a good start to integrating these authentication providers. For many reasons, the project templates default implementation will not work and it is not complete since there were many changes on the provider side now. Also, the DotnetOpenAuth package that is included will not work for Google and requires a different package which we will be using in this article.

Just to understand clearly, we will use an Empty project template and include the Nuget packages ourselves to integrate Google authentication in this article.

Before moving into the implementation, let's understand the application flow to integrate OAuth authentication.

Authentication Flow

1. Users should be redirected to provider to authenticate using provider userid and password. Here, it is Gmail userid and password.
2. On successful authentication, the provider will present a consent screen to allow the third party application (or service provider or relying party) to access the basic user information from the provider. This is one time process, if the user allows access first time then subsequent request will by-pass this step.
3. The provider will call the landing page or redirect page on the client application by posting access token and some default public user details for application to use.
4. The application will create a local authentication ticket (a forms authentication ticket) using the access token and user details the provider passes.

Pre-Requisite

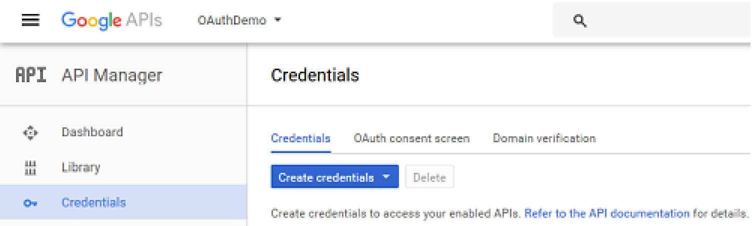
Below things are required to integrate OAuth authentication into our applications. This is a common step for every OAuth provider.

1. Create OAuth Application on provider (Google). Once done, we will get a ClientId and ClientSecret key for using in our application.

2. A Landing page or Redirect page on the application which the provider uses to redirect users after successful authentication. This is where we will authenticate users locally.

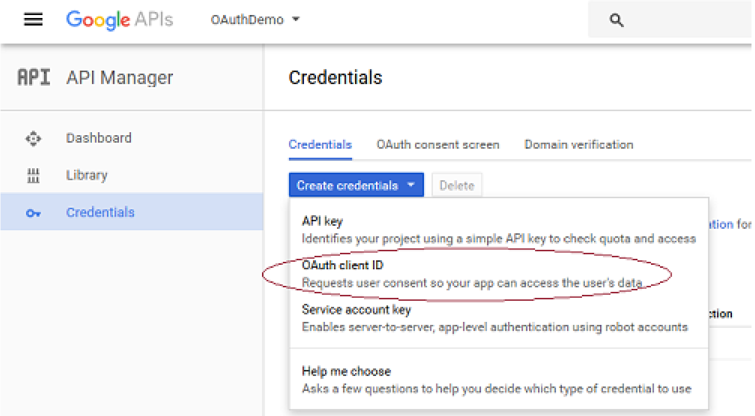
Creating OAuth Application

1. Go to https://console.developers.google.com (https://console.developers.google.com). Sign-in with your gmail id.
2. You will be presented with Google API Manager Dashboard. Click Credentials on the left navigation.



Create credentials to access your enabled APIs. Refer to the API documentation for details.

3. Click Create Credentials > OAuth ClientID.



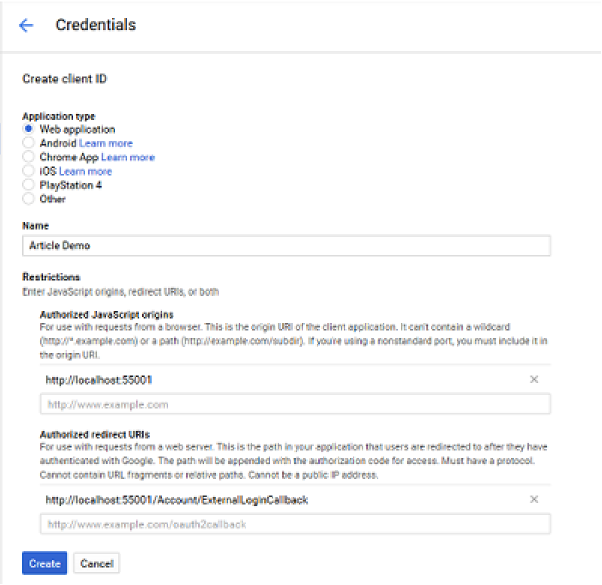
API key
Identifies your project using a simple API key to check quota and access

OAuth client ID
Requests user consent so your app can access the user's data

Service account key
Enables server-to-server, app-level authentication using robot accounts

Help me choose
Asks a few questions to help you decide which type of credential to use

4. Select "Web application" Application type. Type a name for your application. Authorized Javascript Origins is your application root domain. Authorized redirect URIs is the landing page url on your application. You can add multiple urls here. Click Create.



Credentials

Create client ID

Application type
● Web application
○ Android Learn more
○ Chrome App Learn more
○ iOS Learn more
○ PlayStation 4
○ Other

Name
Article Demo

Restrictions
Enter JavaScript origins, redirect URIs, or both

Authorized JavaScript origins
For use with requests from a browser. This is the origin URI of the client application. It can't contain a wildcard (http://*.example.com) or a path (http://example.com/subdir). If you're using a nonstandard port, you must include it in the origin URI.

http://localhost:55001 ×
http://www.example.com

Authorized redirect URIs
For use with requests from a web server. This is the path in your application that users are redirected to after they have authenticated with Google. The path will be appended with the authorization code for access. Must have a protocol. Cannot contain URL fragments or relative paths. Cannot be a public IP address.

http://localhost:55001/Account/ExternalLoginCallback ×
http://www.example.com/oauth2callback

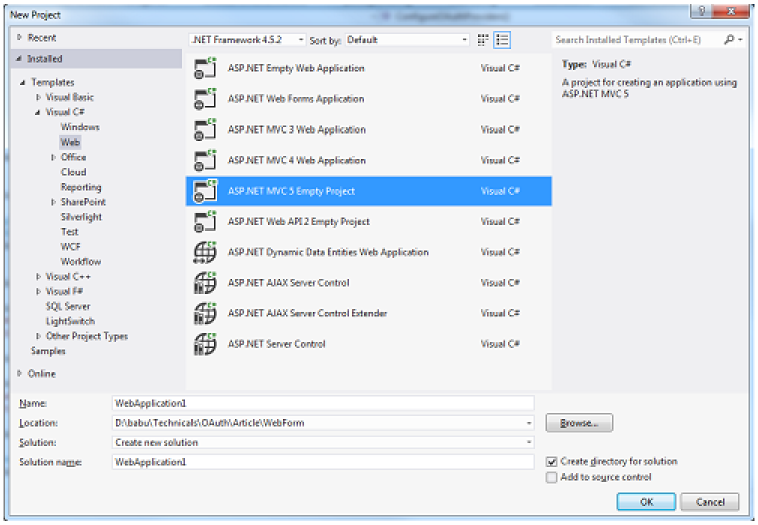
This will create your application and will present the ClientId and Secret in a pop-up. Note down this to configure in the application.

Note – You can also access/edit this details from "Credentials" tab. All created application will be listed here for managing it.

Now, let's create an Asp.Net MVC 5.0 project and integrate Google authentication using the above details.

Creating Visual Studio Project

1. Open Visual Studio (2012/2015) and create a new Asp.Net MVC project. Select Empty project template to add the required Nuget package ourselves. I have used “Asp.Net MVC 5 Empty Project”


New Project
Recent
Installed
Templates
Visual Basic
Visual C#
Windows
Web
Office
Cloud
Reporting
SharePoint
Silverlight
Test
WCF
Workflow
Visual C++
Visual F#
SQL Server
LightSwitch
Other Project Types
Samples
Online
.NET Framework 4.5.2
Sort by: Default
Search Installed Templates (Ctrl+E)
ASP.NET Empty Web Application Visual C#
ASP.NET Web Forms Application Visual C#
ASP.NET MVC 3 Web Application Visual C#
ASP.NET MVC 4 Web Application Visual C#
ASP.NET MVC 5 Empty Project Visual C#
ASP.NET Web API 2 Empty Project Visual C#
ASP.NET Dynamic Data Entities Web Application Visual C#
ASP.NET AJAX Server Control Visual C#
ASP.NET AJAX Server Control Extender Visual C#
ASP.NET Server Control Visual C#
Browse...
OK
Cancel

Type: Visual C#

A project for creating an application using ASP.NET MVC 5

Name: WebApplication1

Location: D:\babu\Technicals\OAuth\Article\WebForm

Solution: Create new solution

Solution name: WebApplication1

☑ Create directory for solution

☐ Add to source control

2. Add the below Nuget packages into your solution.

    1. DotNetOpenAuth.GoogleOAuth2 (The one that comes with default template does not support OAuth 2.0)
    2. Microsoft.AspNet.Membership.OpenAuth
    3. Microsoft.AspNet.Providers.LocalDB

    Note – For simplicity, I will use Microsoft.AspNet.Providers.LocalDB (https://www.nuget.org/packages/Microsoft.AspNet.Providers.LocalDb/) Asp.Net Universal Provider to create and persist user in localDb under App_data. You can call your User data access code to create user once the provider calls the landing page with user details.

3. Add a class file called AuthConfig.cs under App_Start Folder. Let’s configure and add the Google OAuth 2.0 client into the OpenAuth AuthenticationClients collection. This is where we will put the clientId and client secret to connect Google OAuth 2.0 API.

```
public static class AuthConfig
{
    public static void RegisterAuth()
    {
        GoogleOAuth2Client clientGoog = new GoogleOAuth2Client("[Your ClientId]", "[Secret key]");
        IDictionary<string, string> extraData = new Dictionary<string, string>();
        OpenAuth.AuthenticationClients.Add("google", () => clientGoog, extraData);
    }
}
```

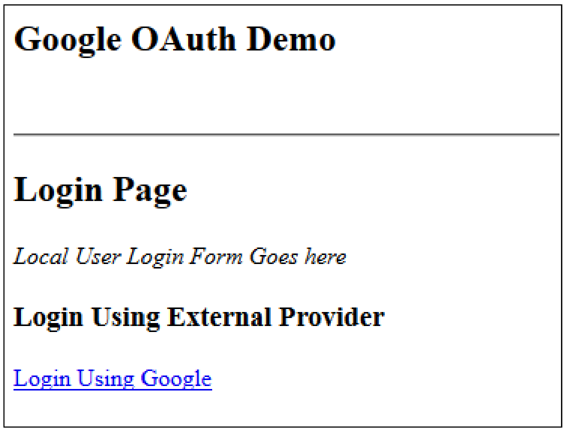
    Include DotNetOpenAuth.GoogleOAuth2 and Microsoft.AspNet.Membership.OpenAuth namespace in using section for the above code to work.

    Note – You actually multiple Authentication clients to this collection like Twitter,FaceBook, etc and access it across the application. For this article, I have added only Google client for demonstration.

4. Call this method from Global.asax Application_Start event.

```
protected void Application_Start()
{
    AreaRegistration.RegisterAllAreas();
    RouteConfig.RegisterRoutes(RouteTable.Routes);
    AuthConfig.RegisterAuth();
}
```

5. Let’s add AccountController into Controller folder and add Login methods. For simplicity, let’s not add local registration and FormsAuthentication local login process here. We will just add a “Login Using Google” button in the login page. So the final page looks like,



Google OAuth Demo

Login Page

Local User Login Form Goes here

Login Using External Provider

Login Using Google

Login view

```
<h3>Login Using External Provider</h3>
@Html.ActionLink("Login Using Google", "RedirectToGoogle")
```

Note – I have added a simple Layout page with simple header text “Google OAuth Demo”.

6. Add an action method to redirect user to Google login URL on click of the above “Login Using Google” link.

```
public ActionResult RedirectToGoogle()
{
    string provider = "google";
    string returnUrl = "";
    return new ExternalLoginResult(provider, Url.Action("ExternalLoginCallback", new { ReturnUrl = returnUrl }));
}
internal class ExternalLoginResult : ActionResult
{
    public ExternalLoginResult(string provider, string returnUrl)
    {
        Provider = provider;
        ReturnUrl = returnUrl;
    }

    public string Provider { get; private set; }
    public string ReturnUrl { get; private set; }

    public override void ExecuteResult(ControllerContext context)
    {
        OpenAuth.RequestAuthentication(Provider, ReturnUrl);
    }
}
```

When user clicks the link, he or she will be redirected to Google login page by specifying “/Account/ExternalLoginCallback” as the redirect URL. This URL should match the URL we saved when creating OAuth application on Google developer console.

7. Next, let’s add the landing page or redirect URL action method “ExternalLoginCallback” to get back access token and user details Google is forwarding.

```
[AllowAnonymous]
public ActionResult ExternalLoginCallback(string returnUrl)
{
    string ProviderName = OpenAuth.GetProviderNameFromCurrentRequest();

    if (ProviderName == null || ProviderName == "")
    {
        NameValueCollection nvs = Request.QueryString;
        if (nvs.Count > 0)
        {
            if (nvs["state"] != null)
            {
                NameValueCollection provideritem = HttpUtility.ParseQueryString(nvs["state"]);
                if (provideritem["__provider__"] != null)
                {
                    ProviderName = provideritem["__provider__"];
                }
            }
        }
    }

    GoogleOAuth2Client.RewriteRequest();

    var redirectUrl = Url.Action("ExternalLoginCallback", new { ReturnUrl = returnUrl });
    var retUrl = returnUrl;
    var authResult = OpenAuth.VerifyAuthentication(redirectUrl);

    if (!authResult.IsSuccessful)
    {
        return Redirect(Url.Action("Account", "Login"));
    }

    // User has logged in with provider successfully
    // Check if user is already registered locally
    //You can call you user data access method to check and create users based on your model
    if (OpenAuth.Login(authResult.Provider, authResult.ProviderUserId, createPersistentCookie: false))
    {
        return Redirect(Url.Action("Index","Home"));
    }

    //Get provider user details
    string ProviderUserId = authResult.ProviderUserId;
    string ProviderUserName = authResult.UserName;

    string Email = null;
    if (Email == null && authResult.ExtraData.ContainsKey("email"))
    {
        Email = authResult.ExtraData["email"];
    }

    if (User.Identity.IsAuthenticated)
    {
        // User is already authenticated, add the external login and redirect to return url
        OpenAuth.AddAccountToExistingUser(ProviderName, ProviderUserId, ProviderUserName, User.Identity.Name);
        return Redirect(Url.Action("Index", "Home"));
    }
    else
    {
        // User is new, save email as username
        string membershipUserName = Email ?? ProviderUserId;
        var createResult = OpenAuth.CreateUser(ProviderName, ProviderUserId, ProviderUserName, membershipUserName);

        if (!createResult.IsSuccessful)
        {
            ViewBag.Message = "User cannot be created";
            return View();
        }
        else
        {
            // User created
            if (OpenAuth.Login(ProviderName, ProviderUserId, createPersistentCookie: false))
            {
                return Redirect(Url.Action("Index", "Home"));
            }
        }
    }
    return View();
}
```

The above code will create user and login locally. On successful login, the user will be created by the provider in localDb under App_Data and it will redirect the user to Home page.

Note *– I have used default membership database to store the user details here. Once you got the user details from google you can call your data access method to store into your application user table and call FormsAuthentication.SetAuthCookie() to authenticate user.*

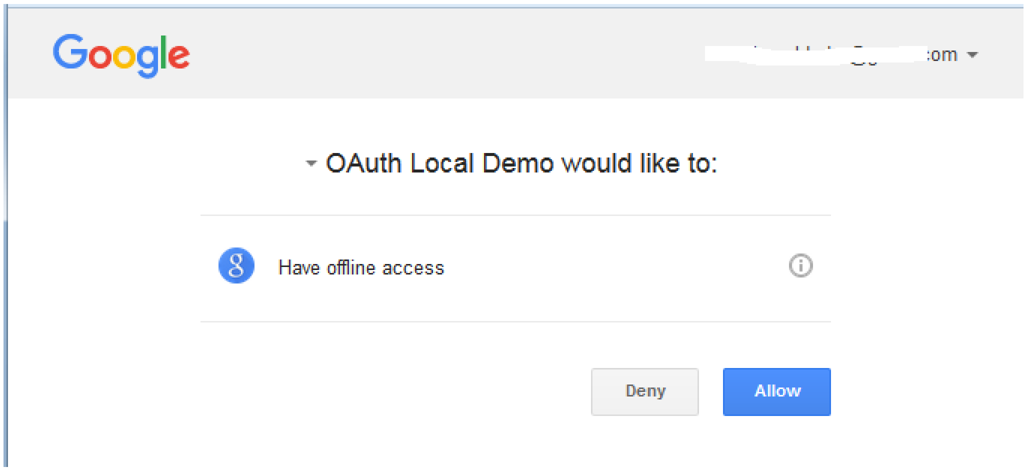
8. Finally, we will add LogOff action to logoff user.

```
public ActionResult LogOff()
{
    FormsAuthentication.SignOut();
```

```
        //Call https://www.google.com/accounts/Logout if you want to logoff at provider
        return Redirect(Url.Action("Index","Home"));
    }
```

9. That's it. Press F5 and see it in action.

On clicking "Login Using Google", we will be redirected to Google login page. After successful login, we will be presented with user consent screen similar to below,



▾ OAuth Local Demo would like to:

Have offline access

Click "Allow" we will be redirected to our application landing page. This would have created user details in localDb database created under App_Data folder. You can view the table using server explorer in Visual Studio.

Download the source attached with this article and see it in action.

I have also included a WebForms implementation which fairly follows the same approach but uses the default project template implementation. Make sure to update the redirect URL on the Google console appropriately before running the application.

Downloads

- MVCGoogleOAuthDemo.zip (/Attachments/2/13/MVCGoogleOAuthDemo.zip)
- WebFormsOAuthGoogleDemo.zip (/Attachments/2/13/WebFormsOAuthGoogleDemo.zip)

Related Read

Streaming Media Contents and VPNs (/posts/60/streaming-media-contents-and-vpns)

The Hidden Benefits Of A Password Manager (/posts/59/the-hidden-benefits-of-a-password-manager)

Make Asp.Net MVC Clientside(jquery.validate.js) Form Validation Error Message Accessible (/posts/58/make-aspnet-mvc-clientside-form-validation-error-message-accessible)

Create Accessible TextArea Character Counter in Asp.Net MVC Using jQuery (/posts/57/create-accessible-textarea-character-counter-in-aspnet-mvc-using-jquery)

Feedback

Title

Title / Subject

Your comment

Your comment....

Posted By

Your Name

Word Verification

Stop air pollution, We need clean air
Type in the box below: BT71

Type Here:

Add Comment

Comments

S
asd
Commented by asd on **9/9/2018 10:34:54 PM**

error al escribir AuthConfig.RegisterAuth(); en Global.asax.cs
Escribir espacio de nombre using nombreproyecto.App_Start
Commented by zac on **7/31/2018 9:36:33 AM**

Error 401
I am getting following error : response_type=code client_id=[Your ClientId] scope=https://www.googleapis.com/auth/userinfo.profile https://www.googleapis.com/auth/userinfo.email redirect_uri=http://localhost:52912/Admin/Account/ExternalLoginCallback state=ReturnUrl=%7B%27web%27%3A%7B%27client_id%27%3A%27595904784846-5dkpf1qpe7rhkj3ldjf36f1kfj3gmg3t.apps.googleusercontent.com%27%2C%27project_id%27%3A%27organic-lacing-203513%27%2C%27auth_uri%27%3A%27https%3A%2
Commented by Altaf on **5/9/2018 12:18:58 AM**

Error
I am getting 401 Error once I login using External login method.
Commented by Shravani on **4/10/2018 10:52:46 AM**

Server Unavailable Exception
While calling OpenAuth.VerifyAuthentication(string returnUrl) it throws The remote server returned an error: (503) Server Unavailable exception at some times. This is not occur for everyone. Some times facing this exception
Commented by Priya on **3/15/2018 2:02:40 AM**

Error

AuthConfig.RegisterAuth(); I'm write this line in application_start in global.asax then show to error .how to solve it?
Commented by Parth jariwala on **2/11/2018 9:27:07 PM**

Code for logout
I would like to comment to provide logout code for google acount. public ActionResult LogOff() { FormsAuthentication.SignOut(); //Call https://www.google.com/accounts/Logout if you want to logoff at provider return Redirect("https://www.google.com/accounts/Logout?continue=https://appengine.google.com/_ah/logout?continue=[your website url]"); }
Commented by Phani Tekkem on **12/19/2017 1:42:01 AM**

Database confusion
Dear Sir I'M new in S-development i need the database details of above authentication system..Please share
Commented by Vishnu Vijayan on **10/19/2017 12:28:28 AM**

New Stories

Top Stories

- Streaming Media Contents and VPNs (/posts/60/streaming-media-contents-and-vpns)
- The Hidden Benefits Of A Password Manager (/posts/59/the-hidden-benefits-of-a-password-manager)
- Make Asp.Net MVC Clientside(jquery.validate.js) Form Validation Error Message Accessible (/posts/58/make-aspnet-mvc-clientside-form-validation-error-message-accessible)
- Create Accessible TextArea Character Counter in Asp.Net MVC Using jQuery (/posts/57/create-accessible-textarea-character-counter-in-aspnet-mvc-using-jquery)
- Accessible RadioButtonList and CheckBoxList in Asp.Net MVC-Section 508 Compliance and ADA Compliance (/posts/56/accessible-radiobuttonlist-and-checkboxlist-in-aspnet-mvc-section-508-compliance-and-ada-compliance)

Latest QuickStart Guides

- What is Web Accessibility Guidelines & Section 508 Compliance? Build Screen Reader Friendly WebSites (/quick-start/18/what-is-web-accessibility-guidelines-section-508-compliance-build-screen-reader-friendly-websites)
- What is Dapper? How to Use Dapper in Asp.Net MVC? (/quick-start/17/what-is-dapper-how-to-use-dapper-in-aspnet-mvc)
- What is REST Services? How to Create REST Services in Asp.Net? (/quick-start/16/what-is-rest-services-how-to-create-rest-services-in-aspnet)
- What is OWASP? What are OWASP Top 10 Security Risks? (/quick-start/15/what-is-owasp-what-are-owasp-top-10-security-risks)
- Using Gulp (gulpfile.js) in Visual Studio 2017 (/quick-start/14/using-of-gulp-gulpfilejs-in-visual-studio-2017)
- What is Gulp? How to Use Gulp? (/quick-start/13/what-is-gulp-how-to-use-gulp)
- What is Entity Framework Code First Migration? A Beginners Guide (/quick-start/12/what-is-entity-framework-code-first-migration-a-beginners-guide)
- What is Dependency Injection? How to use this Pattern in Asp.Net MVC? (/quick-start/11/what-is-dependency-injection-how-to-use-this-pattern-in-aspnet-mvc)
- Learn Asynchronous Programming (async, await) in C# in 10 Minutes (/quick-start/10/learn-asynchronous-programming-async-await-in-c-in-10-minutes)
- What is .Net Standard? How it is different from .Net Core and Portable Class Library? (/quick-start/9/what-is-netstandard)

Browse by category

- **.NetFramework (/AllArticles.aspx?CatID=CAT5)**
- **ASP.Net AJAX (/AllArticles.aspx?CatID=CAT11)**
- **C# (/AllArticles.aspx?CatID=CAT2)**
- **Windows Clustering (/AllArticles.aspx?CatID=CAT9)**
- **Javascript (/AllArticles.aspx?CatID=CAT13)**
- **ADO (/AllArticles.aspx?CatID=CAT10)**
- **Visual Studio (/AllArticles.aspx?CatID=CAT15)**
- **Webservices (/AllArticles.aspx?CatID=CAT12)**
- **SqlServer (/AllArticles.aspx?CatID=CAT6)**
- **Reporting Services (/AllArticles.aspx?CatID=CAT7)**
- **Directory Services (/AllArticles.aspx?CatID=CAT8)**
- **IIS (/AllArticles.aspx?CatID=CAT4)**
- **ASP.Net MVC (/AllArticles.aspx?CatID=CAT16)**
- **Sharepoint Services (/AllArticles.aspx?CatID=CAT14)**
- **jQuery (/AllArticles.aspx?CatID=CAT17)**

Contact Us (/home/contact) | Disclaimer (/home/disclaimer) | Terms and Condition (/home/termsandcondition) | Privacy Policy (/home/privacypolicy)